JULY 1992 £1.95

SINCLAIR QL WORLD

2 7-10
11 13 14
18 22 25
30 33 SLOT CARD
34 40 DISC IC
42 44

DIY TOOLKIT: WINDOWS FOR QDOS

MEETING IN MÜNSTER

LAST NEW QL's READER OFFER!

DIRECT TYPING FROM ARCHIVE

9 770951 933016 07

SINCLAIR QL WORLD

Editor
Helen Armstrong

Publisher
Mark Kasprowicz

Advertising Manager
Jim Peskett

Creative Director
John Stanley

Designer
Steve Billington

Magazine Services
Pauline Wakeling
Frances Maxwell
Linda Miller

Sinclair QL World,
Published By Arcwind Ltd.
The Blue Barn,
Wootton, Woodstock,
Oxon. OX7 1HA
Tel: 0993 811181
Fax: 0993 811481
ISSN 026806X

If you have any comments or difficulties please write to the Editor and we will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies.
Back issues are available from the publisher price £2.50 UK, £2.99 Europe. Overseas rates on request.

Subscriptions from: Arcwind
The Blue Barn, Wootton, Woodstock, Oxon OX7 1HA
UK: £23.40
Europe: £32.90
Rest of World: £40.90

Reprographic Services: The Wace Group, London.
Printed and Bound by
William Gibbons, Planetary Road, Willenhall, West Midlands
Distributed by: IPC Market force, King's Reach Tower, Stamford St, London SE1 9LS.

© 1992 Arcwind Ltd,
Sinclair QL world is published 12 times a year. All rights reserved. Reproduction in whole or in part without written permission is strictly prohibited. We welcome contributions but can not be responsible for any unsolicited contributions, all material must be supplied with a SAE.

CONTENTS

JULY1992

4 TROUBLESHOOTER ● Updates from all over
11 QL SCENE ● Miracle hands over the Trump Card
13 QL SCENE ● Cowo expand their list
14 OPEN CHANNEL ● Thoughts on the QL's future
16 SOFTWARE FILE ● Polytext
18 JUST FOR STARTERS ● Basic database part 3
22 ONLY FOLLOWING ORDERS ● Regular payments with Abacus
25 NEW USER GUIDE ● IDEC$ to INPUT
29 QL SCENE ● Digital Precision updates
30 ARCHIVE ANSWERS ● Direct typing
32 SOFTWARE FILE ● Rockfall
34 DIY TOOLKIT ● Windows for the QL
38 SOFTWARE FILE ● Pudge
40 SOFTWARE FILE ● QuickDisk and DiskTool
42 QL INTERNATIONAL ● A Munster event for users
43 CLUB ACCESS
44 SYSTEMATIC MACHINE CODE PROGRAMMING ● Part 8
50 READERS' OFFER ● From EEC Ltd.

COMING SOON!

We have an updated version of Cowo Electronics' QL and Thor front end QTop; several new games reviews; a publisher working on suggestions for would-be software authors when sending their wares for assessment; more meetings reports, and an interesting new Mouse from EEC Ltd.

TROUBLE

A PROBLE

Bryan Davies has been trying some of the new upgrades from several QL suppliers this spring.

You will probably by now have read the front page of *QL World*, and realised that the magazine is under new management. The company uses Apple Macintoshes for their magazine layout work, and can load files from standard PC programs into the Macs. Had the QL been developed further by Sinclair, it might by now have been quite like a Mac, with a more-powerful 68000-series cpu, hard disk drives, better graphics display, a WIMP user interface, and a much more comprehensive operating system. Thanks to various suppliers, steps have been taken to update the QL in some areas; we have a faster processor, in the form of the Gold Card, and a WIMP interface, *QPac*. Improvements to the Qdos operating system have been piecemeal, and leave a lot to be desired by the average user. The graphics haven't really been tackled yet. The story is far from ended, though. There is no shortage of QL material, circulation of QL World and membership of Quanta both appear to have stabilised and could be increased somewhat, so there is good reason to think our computer will continue to be well-supported as it comes up to its tenth year.

New products

Jochen Merz was demonstrating the QL emulator running on an Atari said to have a 33 MHz 68020 cpu at the Bristol Quanta meeting. Not surprisingly, it was reported to be quite fast! Miracle Systems are working on both higher- and lower-specification versions of the Gold Card. Now we know just how usable QL software is with the 16 MHz 68000 Gold Card, the thought of what will be possible with the 68020 or 68040 cpu is truly exciting. Digital Precision has a speed demonstration which should register with most people, and it makes use of the *Perfection* wordprocessor running on a standard Gold Card system. A single file containing about 40% of the bible (roughly 1.6 MB!) is loaded into Perfection, and Copy and Spell-check operations run; the times taken are in the low seconds, not the minutes (or even hours) that some people might expect. Beta-test versions of Perfection are doing the rounds again, and they contain some significant improvements, so be prepared for a new release shortly.

Faster emulation

Not new, but due out in updated form any time now, is Digital Precision's *Conqueror* PC emulator. Amongst the new features are support for expansion memory, which can be well-utilised by the DR-DOS 6.0 (or 6.1) operating system that is supplied with the Special Edition version. The PC's conventional memory limit of 640 KB was raised to 667 KB by the original version of Conqueror, and has now been jacked-up further to a reported 736 KB. Having been struggling to get an extra few bytes' working space on my own PC recently, an extra couple of hundred kilobytes seems a dream, but Conqueror provides it. Naturally, use of the Gold Card increases speed of operation considerably, but it would be losing much of the GC's advantage if its extra memory could not be used also, and Conqueror can now do just that, making a fully-usable PC out of the QL.

Another significant improvement is that the program is tuned for operation with the Gold Card and extra-high density disk drives. DR-DOS 6.0 is up-to-the-minute PC software, and merits some description here. The basic 640 KB which is the nominal PC and DOS 'ration' has been exceeded in several ways for some years now, but many PCs are still restricted to this amount; DOS 6.0 enables several chunks of the operational software to be loaded into areas of memory away from the basic 640 KB. The extra memory provided by the Gold Card can also be used as extended memory, and this type of memory is vital to the operation of that masterpiece of seductive programming, Windows. It also enables ramdisks to be created. One major feature of DR-DOS 6.0 is its disk cache routine, which can also make use of the extra memory, to give a considerable speed-up of disk read/write operations. The SuperStor feature uses on-the-fly file-compression to increase the available hard disk space by a factor of two. If you haven't got a hard disk, but have an ED drive, you can create a DOS area of about 6.4 MB on it. That is near hard disk capacity.

Try a rom

The version 2.28 rom in the Gold Card is reported to sort out various disk-associated problems. As the reports are all second-hand, please don't take them as being a guaranteed cure for any disk problems you may have, but it is worth contacting Miracle Systems, if you have trouble you think is related to the Gold Card, to ask their advice on whether or not the 2.28 rom is needed in your GC. The reports suggest this rom deals with file/directory errors on ED disks, incompatibility with one-third-height Mitsubishi 3.5in drives, failure to boot of *QL Invoicer* (one drive being ignored), and unspecified trouble with *Assembler Workbench*. As disk-corruption with some ED drives was definitely established, and hasn't occurred on the drives in question since the V2.28 rom was fitted, there may be reason to hope corruption reported on DD and HD disks could be cured by this rom also.

Miracle state that "(the V2.28 rom) fixes the directory corruption problem and also allows the old one-third-height Mitsubishis to be used". The new rom has been tried in my JS and JM systems; unfortunately, it has made the Mitsubishi drives unavailable, so can't be counted a success, for me. Miracle have been asked to look into the problem some more. With the recent drop in price of the GC, Miracle have dropped the Trump Card from their product line, but another, smaller dealer will be selling and supporting it, with Miracle's assistance.

Text87 Plus4

I waited in vain for a long time for a copy of *text87 Plus-4*. It wasn't hard to guess that a main reason for the delay was the late discovery of bugs, but it is not exactly unusual for programs to be shipped with bugs in them, and beta testers certainly don't expect their copies to be free of problems. There is an old adage that you are only successful if you get noticed, and being caught 'with your trousers down' is not necessarily going to hinder your ca-

SHOOTER

M S O L V E D

reer — to the contrary, it may well improve your prospects. Looking at the micro scene, it would not be hard to name more than a dozen major programs in world markets that were released when they were scarcely out of the factory test stage, yet were very successful in sales terms. *(Very true, Bryan, and look at all the woe and sorrow that has caused the users.)*

What was all that about? Mainly that I have been unable to say anything much about text87 - a program I have made much favourable comment on in the past - other than that I didn't have the latest version. A copy of Plus-4 2.0 finally got to me in mid-April, and another arrived early in May, together with a revised driver for the Epson GQ-5000 laser printer. Initial experience with it shows it to be much-improved over the previous generations of text87, with considerable expansion of the facilities. There are now windows for different documents, or different views of the same document, a print preview, simplified setting-up of typestyles and attachments, and various other functions.

The menu structure may still be rather daunting for users who have experienced only *Quill*, but commands are straightforward to access, with full use being made of the now-standard method of keying the first letter of the command required (after F3 has been keyed). Inevitably, there are some cases where another letter has to be used, but this isn't a real problem. If the user doesn't investigate the functions which take text87 far beyond Quill in capability, there should be no difficulties with the keyings, as they are Quill-like. The current version appears stable, suggesting that the bugs in early versions which caused crashes or lock-ups have been fixed. A major feature of text87 has always been the 'real' screen presentation of text, and the ability to print from software founts supplied by *fountext88*.

To users who value what comes out on paper, text87 offers an unbeatable range of founts. If you have a Hewlett-Packard DeskJet with added founts, you can use the *typeset90-DeskJet* drivers to provide print with founts every bit as professional as achieved from current PC programs. The samples provided in the information on new drivers look just the same as ones from a recently-released PC program; using the same printer, you wouldn't be able to tell which computer and program were used.

Another Special Edition

Not to be outdone, Digital Precision are now sending-out beta-test versions of *Perfection Special Edition*. This version addresses comments made by various users. One such is the ability to move from the first character of the current line to the last one of the previous line by pressing the left-cursor key once, rather than moving the cursor up one line and then using additional keying to move to the end of the line. A small thing, maybe, but one which can transform the program for users (like me) who frequently perform this operation. More important for other users is the increased number of Strips, which will permit more printer functions to be called from within documents (rather than having to be 'hard-wired' in the printer-driver). There are also far more Translate entries for printer-drivers - enough for all but the most way-out requirements. Re-formatting has caused some aggravation, being a (comparatively) slow process in the past. It has been speeded-up considerably and, in the process, Search and Replace operations have received a boost, too.

On the subject of printing, DP has sent in a sample of print from *Professional Publisher* on a HP LaserJet II, and the quality is as you would expect from that printer - considerably better than from your favourite 9-pin DMP.

More on disks

Those who buy lots of HD 1.44 MB disks should be interested to know that the supplier Mediaware (see INFORMATION box) appears to have dropped some of the pretence associated with selling unbranded disks, and may be supplying branded ones even when unbranded are ordered. It may have been a temporary measure, when filling my order, however, so please don't take this to mean it is policy on the part of this supplier. Specifically, I ordered 50 unbranded ('bulk') HD disks, at a price of 68p plus VAT each; what I got were labelled 'Precision' brand, made by Dysan, one of the main disk manufacturers. Post and packing raised the total price by 1.50 plus VAT, but it covered other disks and two disk boxes, so that the effective price per HD disk was about 80p including VAT. While I generally recommend buying disks without any brand labelling on them, I don't object to branded disks for the same price!

News from Italy

The Italian QL user group, QItaly, has sent disk number 20, which contains the club magazine for September-October 1991. As usual, this video magazine is well-presented. The screen shot shown gives details of membership for 1992. Note that the magazine makes use of other founts and sizes besides the standard QL ones. There are some good colour screen shots taken from new or upgraded programs, details of QL meetings, conversations, correspondence, etc, and articles on subjects of current interest, such as the 3.2 MB ED disk drives from Miracle. Most of the material

QITALY CLUB

CLUB ITALIANO PER UTENTI DI SINCLAIR QL

Per informazioni:

Roberto Orlandi
Via Brescia 26
25039 Travagliato (BS)
Tel.(030)6863311 0

Eros Forenzi
Via Valeriana 44
23010 Berbenno (SO)
Tel.(0342)492323

Quote iscrizione 1992:
L.25000 quota associativa
L.25000 quota integrativa per ricevere QITALY Magazine

Cosa offre il Club:

1) 4 numeri annui di QITALY Magazine
2) 2 numeri di MONDO QL, rivista su carta
3) Consulenze per problemi hardware e software, acquisti di gruppo a prezzi ridotti, programmi PD da tutto il mondo, manuali in italiano, programmi originali ..... e molto altro ancora!

is, naturally, in Italian, but translations are provided for usage instructions, and a fair guess can be made at what is being said elsewhere if you know the subject matter of the text (for example, a letter of mine to Eros Forenzi!)

Still going out

Despite the fact that the QL production line came to a standstill some years ago, new QLs continue to be sold in significant quantity. One way and another, complete boxed QLs have been seeing the light of day and look set to carry on doing so for another year or so. EEC Ltd. confirm that their sales of new QLs run into four figures, and that they still have a reasonable stock of them. This company has become the main supplier of complete systems, and we receive almost no complaints about it (the only complaint received by me arose from what appeared to be a genuine mail problem, and it was sorted out). In the event of the supply drying up, EEC may have to turn to selling Uncle Clive's new lightweight bicycles to maintain their link with Sinclair!

Readers' letters

Brian Kemmett reports receiving a copy of Professional Publisher from **TK Computerware** without the instruction booklet. TK say they have sent two copies of the booklet since, and re-sent one of them after it had been returned by the Post Office as 'not collected'. As the booklets appear to have been sent Recorded Delivery, and the program disk arrived safely, the Post Office has been contacted by both parties, but the reported responses are not in agreement. Meanwhile, Kemmett cannot use ProPub and has contacted the Trading Standards Office; he has also threatened legal action.

A letter concerning complaints by readers **DR Scott, C McKay and AP Campbell**, sent in mid-March to TK, appears to have gone astray and no reply has been received to it or to a verbal request for comment on their complaints. However, Scott has since written to say he has received the QL Scanner that was the subject of his complaint. **Keith Johnston** listed calls and letters that he has sent concerning a Z88 Centronics interface, ordered on 10th December 1991 from TK and not received as of 3rd April. As with other complainants, his experience is of not getting replies to his 'phone calls and of being told that the goods have been lost in the post. He contacted the Royal Mail Customer Care Unit, and the 'Raw Deal' help section of the Sunday Post newspaper, but both stated that they had been unable to get a reply from TK. The VISA credit card company pointed out that refunds are not given unless the value of the goods purchased totals more than 100.

Both **Digital Precision** and **Software87** say they have cleared the backlogs of orders which occurred at the start of the year, and the lack of further letters of complaint appears to confirm that all is now back in order.

B.D. Custerson sent a 3.5in 720 KB floppy disk, with the message 'Will not format. For discard use' on it. This was in response to my comments on problems with floppy disks and drives. Sure enough, the disk showed no sign of a format, in either QL or PC. Attempting to format it in a QL ED drive produced the message 'format failed', but that was not a conclusive indication that the disk was unformattable. Next step was to try a 720 KB DR-DOS format, and that was successful, with no errors recorded. The same result was obtained from a 1.44 MB MS-DOS format. Obviously, a check with a QL DD drive was needed, so another QL was switched on and attempts at a format made in both a DD and an ED drive; two failures were recorded in the DD, but a successful format, with no errors, was obtained first time, in the same ED drive that had previously recorded a failure. In case the presence of a four-drive adapter, and the ED drives, was influencing results, the dual DD drives were plugged-into the second QL on their own; result, a no-error DD format first try.

Back to the first QL, with only the DD drives connected. This time the result was quite unexpected - an error-free 360 KB format! What was different? The first QL system had a JS rom and Mk.2 Gold Card, whereas the second one had a JM and a Mk.1 Trump Card. A four-drive adapter was being used with both QLs, to allow dual ED and dual DD drives to be used together. Finally, the rom in the GC was changed from the 2.28 back to the 2.25 version, and the disk was then formatted successfully, to 720 KB, in both DD and ED drives.

My experience in the past has always been that the QL will almost invariably format without errors disks which give errors on the PC. This instance appeared to be a reversal of the usual situation, but that was very likely a function of the GC rom; Mr Custerson didn't give any details of his system, but the disk is on its way back to him, to see how it behaves now it is formatted, and accompanied by a request for details of his system configuration. Another reader with disk trouble is **J. Paul Bissonnette**, who lives in Otterfing, Germany. His bad disks were branded Sony and had been in use for four or five years. He is quite convinced that the problem lies in the disk media, not in any external influences such as nearby magnetic fields, misaligned drive heads etc. Regardless of the type of computer used when formatting these disks, bad sectors appeared after about a week, even if the disks were not used. It would have been interesting to look at the disks but, not surprisingly, they were consigned to the rubbish bin.

Life of Leisure

P.H. Tanner has been a constant correspondent for some time, despite being told that my knowledge of programming was quite insufficient to answer some of his queries. His latest letter makes a non-technical point which should ring a bell with many QL users. How many users have discovered that retirement from work doesn't seem to produce any 'free time'? Do you wonder now how you ever found time to go to work at all? Certainly, computing has to be one of the best gap-fillers around; my own experience has been that almost everything else has gone by the board, to make way for the increasing demands of the keyboards. Despite having a diverse background in computing, Mr Tanner resolutely steers clear of some of the devices others regard as essential; for example, he has no disk drives. He still uses Quill for his wordprocessing. However, he has two QLs more-or-less permanently linked together and passing information between them throughout the day. In the PC world, the talk is increasingly of networked computers, on-line to limitless sources of information around the world and constantly passing 'the word' backwards and forwards. Much of what is said can best be described, politely, as rhubarb, but there is undoubtedly some good use for networking and it won't be long before computers are supplied complete with all the necessary devices for linking into networks - modems, fax cards, CD rom drives etc.

INFORMATION

Disks and boxes:
Mediaware Ltd.
Unit 9
Huntsbank Industrial Estate
Crewe Road
Wistaston
Crewe
Cheshire CW2 6QT.
Tel. 0270 666616
Fax 0270 665556

New QLs:
EEC Ltd.
18-21 Misbourne House
Chiltern Hill
Chalfont St. Peter
Bucks SL9 9UE.
Tel. 0753 888866
Fax 0753 887149

QL SCENE

Miracle hands Trump Card over to Qubbesoft

Miracle Systems have licensed production and support of their popular and long-standing memory expansion units, the **Expanderam** and the **Trump Card**, to Ron Dunnett of Qubbesoft.

Said Stewart Honeyball of Miracle Systems: "The time we were taking on administration and so on was taking attention away from designing new stuff, and we decided that it was not really sensible for us to carry on doing it. Ron is in a better position to do it, and we thought that it was far better to hand it over to somebody else than to ditch it." Support both for existing Trumpcard and Expanderam users will be through Ron in the first instance, although, Stewart says, "We will be working closely with Ron."

This move will come as a surprise to Miracle customers who have known the Trump Card as a popular and long-standing item on Miracle's list. It reflects in part a situation in which the Trump Card and Miracle's Gold Card were effectively in competition with each other, at a time when Miracle wish to devote their resources to developing and stretching the QL hardware base. Ron Dunnett himself has considerable hardware experience, particularly in the area of disk drives, and has recently taken his redundancy from a long-standing career as an opportunity to devote time to his QL hardware and software business.

The Trump Card will now retail at the slightly lower price of £95, the 512K Expanderam at £45, and the zero-ram version of the Expanderam at £20.

An item which is staying firmly on the Miracle product line is the recently-redesigned Centronics serial to parallel converter. All the electronics for this classic items are now contained in the metal hood of the Centronics plug - "It's pretty neat," says Stewart. The price is now down to £25 all inclusive.

On the Qubbesoft front, Ron adds that the C68 C Compiler is now up to version 3.00, updated with a 64-bit floating point and other advances. The new larger source code is not yet available, as it is being archived to two disks "rather than increase the number of disks".

Contact Ron Dunnett at **Qubbesoft, 38 Brunwin Road, Rayne, Braintree, Essex CM7 5BU. Tel. 0376 47852.**

Digitiser Now Gold Card compatible

The CQV I real time video digitiser manufactured by CL Systems is now compatible with the Miracle Systems Gold Card. Hardware and software modifications have been made to handle the increased processing power of the Gold Card. The speed increase measured with the Gold Card in use is more than 2.5 times, translating to a picture update time of 12.5 frames per second in quarter screen, and 7 frames per second in full screen mode. CQV1 users with the Gold Card can now achieve near-full-live-motion video on the QL screen.

CL Systems is offering an upgrade to current users who return their old QV1. The unit will be modified and returned with the upgraded Imaged software. The upgrade cost is £30 for UK users and £33 for overseas users, inclusive of all carriage.

Orders and enquiries to **CL Systems, 403 Chapter Road, Dollis Hill, London NW2 5NG. Tel/fax 081 459 1351.**

NASA bug warning

NASA Computing in Norway has a message for customers who bought the new file manager disk Mate at the Munster show: a bug which can damage date when the 'Write dir' command is used has been fixed. The bug arose from a fix on an earlier bug in the development process. Users should return their original disks for the free upgrade to:

NASA Computing, Nerheim, N-5580 Olen, Norway.

Back Issues

There have been a considerable number of enquiries about back issues of QL World recently. We are still trying to find out where older back issues are available, and we will publish a guide to what is available and how to obtain it shortly.

Upgrades and new material from Merz

Jochen Merz Software has released QD 4.00 at the Munster workshop. New features of the program include GOTO a list of SuperBasic procedures, functions or assembler labels; improved search/replace menu with four different search strings, and others. An upgrade for current users, with a new manual, costs £10 when the current disk is returned.

A new game running under the Pointer Environment is *The Lonely Joker*. This combines three different patience games (similar to those seen on some versions of Microsoft Windows), which can be played with or without the mouse. The game costs £14.90, and a ram extension and SuperToolkit is required.

Merz showed Qdos running on the Atari-ST QL Emulator with a 68030 card running at 32 MHz, with programs running approximately 1 to 20 times faster than an unexpanded QL. The software for 68000/10/20/30/40 versions will be available after April. The 68030 accelerator board costs £500 for a 25 MHz version and £600 for a 32 MHz version, and will fit every Mega ST.

For users lacking an 1005-compatible ACSI/SCSI adaptor, the new Lacom adaptor from Merz can be used to connect SCSI drives to the Atari ST. Small and convenient at 11 by cm, it costs £66.

The QL Emulator itself can now be bought for only £139.

Another new product is the QL SER Mouse, a mouse-driver for serial IBM-style mice to one of the QL's serial ports. It behaves in a similar way to the Qimi mouse, but has three buttons, with ESC in the middle, WAKE to middle+left, and SLEEP to middle+right. The driver costs £13.90.

Contact **Jochen Merz Software at Im stillen Winkel 12, 4100 Duisberg 11, West Germany. Tel. (local) 02 03 501274.**

QL SCENE

COWO expand their range

Cowo Electronic, publishers of QTop, have expanded their range of QL products, alongside its SuperQL hardware project. They have brought back to the market the text editor ArcEd, formerly sold by the now apparently-defunct Dansoft. The program has been revised and updated, and a fee update is available to current users on return of the original disk, self-addressed envelope and return postage.

The SuperQL project has been codenamed ExeQtor. The first tower-mounted machines were delivered in December 1991 and 12 were spoken for by the end of March. Cowo is planning a limited edition of up to 50 units.

Cowo is now the main, if not the only commercial, support for the Thor range, and they have an extensive list of products for the Thor, including Argos rom upgrades, mouse and eprom programming service. Users with Thor enquiries are asked to quote their model and specification; serial number; Qdos version number; SuperBasic version (use PRINT VERS); Argos version; IO rom version (Thor XVI); XChange version; total ram size; keyboard type (language, 84 or 102 keys); whether the SCSI port is installed and if it works; manual type (version and language) and any other relevant information.

Orders and enquiries to **Cowo Electronic, Urs Konig, Munsterstrasse 4, CH-6210 Sursee, Switzerland.**

DIYT correction

There was an error in the *DIY Toolkit* _DEF project published in January and February's *QL World*. Graham Worsnop of Surrey has spotted that the table at the end of the code lacks a terminal zero word. This could cause CALL to lock the machine, and has been corrected in the current DIY disk version.

There is no problem if the code is loaded onto the common heap with *Turbo Toolkit*'s ALLOCATION or the DIY Toolkit extensions LINKUP and RESERVE, as zero bytes are automatically added to pad the space to an eight-byte boundary. The same is true if RESPR is used on Minerva while tasks are running.

However the extra zero word must be added before loading the code into resident procedure space, or merging it with other toolkit files.

To correct the hex loader, change the last line to read thus:

```
1010 DATA
'46250000','*',52617
```

If you have saved the code file but not the loader, this sequence of commands will load the old file, link it correctly to Basic, and save the corrected version:

```
X=RESPR(628)
LBYTES FLP1_DEF_CODE,X
POKE_W X+626,0
CALL X
SBYTES
FLP1_NEW_DEF_CODE,X,628
```

To fix the assembler source, add DC.W 0 just before the end. Simon Goodwin apologises for the error and will try never to make the same mistake again!

Machine Code Errata

One line faded out of Listing two in Systematic Machine Code Programming last month (part 7). The faded line should read:

```
CMPI.B BYTE,A1 ;BYTE
ADDRESS IN A1
```

Let us know if any other lines were faded in your copy.

QL Systems WANTED

QL user Brian Richardson of Wallington is hoping to gather together a QL system for the use and assistance of residents at the Harvest Care Homes and school for the elderly and mentally disabled, which QL World understands has been affected by government legisation withdrawing support.

Mr. Richardson says that he "grew up with Sir Clive, metaphorically speaking, so I am used to computers he designed up to and including the QL." What he needs to start his project is donations of working QL hardware. He says: "Anything will be of use, broken, smashed, or in (hopefully) perfect working orders; expansions, disk drives, monitors, printers, software, anything QL-related at all, even single microcassettes or games."

The residents of the home have learning difficulties, but, says he, "You never know, one of them may turn out to be the computer genius of the future!" Who needs genius? Anyone who can offer anything to help these people to become regular computer users, like the rest of us, please contact **Brian Richardson at 21, Firefly Close, Roundshaw Estate, Wallington, Surrey SM6 9HE.**

Text87 Plus4

Text87 Plus4 is now available from Software87. Plus4 is a bottom-up re-write of the established wordprocessor text87, best known for its high-quality multiple-font handling.

Among other accomplishments, the new text87 offers an enhanced menu system with context-sensitive information; new dialogue boxes; cursor move by character, word, line, paragraph, screen and page; erase by character, word, line and block; block and goto operations; very fast search and replace in both directions, case dependent and independent; hard space and hard and soft hyphenisation. File operations include merge, block save as plain Ascii or fully formatted document, import (*Quill* files retain style markers) including *Archive* and *Abacus* files; spellchecker with dictionary browse and automatic replacement supplied as standard, with size and language options, including American within the large English dictionary; up to eight documents open at once, with more than one window available per document; page preview.

Continuing its tradition of font handling, text87 Plus4 can use 'any combination of small and large founts ... multiple paragraph formats with different line-spacing for each ... several columns plus headers and footers ... used with the appropriate printer-driver, text87 Plus4 can utilise the founts and character sizes built into modern printers.'

Text87 Plus4 also focuses on fast document handling. Times of 5 seconds to reformat a 385-line document (after a margin reset), and 5 seconds to perform a 58-instance search and replace are quoted. Plus four is designed to be fully compatible with all QL roms including US roms, the Trump Card, Gold Card, QL/Atari emulator, *QPac II* and *Taskmaster. Toolkit II* Altkeys can be Hotkeyed in the Pointer Environment.

Text87 Plus4 costs £79. 2488 24-pin drivers cost £15, typeset90-deskjet drivers for HP Deskjets £20; same for GQ3500/5000 and EPL4100/7100 £40; fountext88 and founted89 for 9- and 24-pin printers £40. Existing users should enquire for an upgrade price. Prices include all carriage including airmail overseas. Orders (cheque, Eurocheque, postal orders) and enquiries to **Software87, 33 Savernake Road, London NW3 2JU, UK.**

OPEN CHANNEL

Open Channel is where you have the opportunity to voice your opinions in *Sinclair QL World*. Whether you want to ask for help with a technical problem, provide somebody with the answer, or just sound off about something which bothers you, write to:
Open Channel, Sinclair QL World, The Coach House, Tackley, Oxon OX5 3BN.

Support

I have started to write letters on the subject of the QL's future a number of times. Each time, my thoughts have changed, and the letter has not been finished. I have been working for a large publishing software house (that does not support the QL!) for a couple of months now, and the company attitude to hardware and software (we use Macs, PCs, Amigas and workstations among other things) has helped stabilise my ideas.

After all the pie-in-the-sky comments on the subject, hopefully the following represents a down to earth view.

1) Emulation - in other words, trying to perpetuate the QL scene through another machine's success. To my mind, whether the QL is pretending to be a PC (or Amiga, or whatever), or whether we provide another machine with ways to run QL software, the result is much the same: actually having a machine is better than emulating it. Running software on a machine in its native mode is far more efficient than emulating it. The best emulation is comparatively inefficient. In any case, there is no way you can emulate a Mac.

2) Total rebuild - redesigning the QL to be utterly desirable and selling millions. This is unlikely. Look at the manufacturers who pour millions of pounds into new machines and operating system upgrades. Unless a complete 'new QL' could find some previously hidden niche market, its sales would never justify the development costs that would be incurred.

3) Upgrades - or, why change a winning formula? Why not? They work, they keep current users using the machine, have lower development costs, and provide an incremental upgrade path. Not only that, but there are well-established QL suppliers - who are expert at producing well-considered, marketable upgrades - who have kept the machine going as long as it has already.

The point of my letter is: don't waste time dreaming of transporting the QL to some mythical glory. Instead, support the likes of Miracle Systems, T F Services and Digital Precision, and so on. In return, they are likely to do everything they can to keep the QL strong. If you want to see a 'Super QL' - buy their products, don't pirate software, and tell them what you would want to buy in the future. They are capable of doing the rest.

Andrew Toone
Liverpool

Editor's comment: the main problem encountered by the QL over the years is that the hardware has been somewhat unreliable and indiosyncratic compared to, say, PC hardware. I say 'somewhat' advisedly; anyone who uses a PC compat for other than basic wordprocessing or spreadsheeting will know that the frustration and timewasting of getting hardware and software running easily matches that of other formats. We have now come to a point where long experience is producing consistently efficient upgrade paths - often at first or second stab, which is a good record. In other words, Andrew is right.

Editor's notebook

To my great joy, everybody by now should have their first copy of the new QL World, with our new address, new phone number, new information on subscriptions, and explanation for our recent idiosyncratic scheduling. No more waiting for post lying lost at Panini House. No more wrestling with subscription cheques made out to HHL (or even Maxwell).

Then there is the new typesetting system. New to us, that is: we would like to get a line from anyone who has experience of porting text straight from a QL system to a Mac system. I am also collecting ways of converting QL text files for the PC-based part of the system. The most successful route so far uses PC Conqueror, which converted a text file, including assembly language listings, losing nothing but the tab markers.

Richard Alexander of CGH comments that they publish a program called 3D Terrain as well, and that it is not the same as the one from Sharp's that we reviewed last month.

Stuck

Somebody may be able to help me with this. I am using Quill to write my letters, and I have a problem with the Printer_Dat. At the end of the letter, the printer head stays at the same position that it stopped in. When the next letter is printed, the carriage return does not work and the printing starts in the middle of the page.

The only way that I can get over this is to switch the printer off and on again, so that it resets.

The printer I am using is a Taxan, and has been installed correctly as far as I know, using the information in QL User, September 1985.

Carry on with the good work.

A R Kempton
Southampton

Thanks

Thank you so much for sending replacements for the two issues of QL World which did not come: Lazarhold have sent us a copy of the January 92 issue, but that was all. I am so pleased that we once more have an unbroken sequence from the May 1986 issue, which was bought just after we got our first QL.

Since then, most of what I know about computers generally, and the QL in particular, has been gleaned from QL World: my wife relies on me for anything 'technical' (the blind leading the blind?) as she uses the QL mainly for WP and family history research.

Having finally settled back here from Gibralter, we are waiting for the arrival from there of our household stuff, including the second QL. Then we'll have one apiece, and no more competing for 'hands on' time!

Many thanks for all the interest and accurate information: we do appreciate it very much.

Gordon and Dorothy Perrett
Weymouth
Dorset

Abacus

Abacus is much under-rated. I am often surprised how much programming is possible in a single cell. More articles on Abacus, please. Does anyone know how to Order in reverse, ie largest to smallest? Most other spreadsheets can order in either direction.

Among the spreadsheets I use are monthly bank statement reconciliations, which I keep for a couple of years. An update program I have written automatically deletes reconciliation sheets over two years old. This depends on each file having a recognisable name which indicates teh date, and is not more than eight digits long. I always put the file name in call A1, and for the reconciliation sheets I use a form 'Chb9203' (as for March 1992). I include in cell G1 the number of the month which is used at various points on the sheet. Increasing this number produces the next Monthly sheet. In order to produce a new filename in cell A1 at the same time, I coded in the following:

```
"Chb"+date(0)(3 to 4)+if(G1<10,("0"+chr(index(col()+6,row()+48)),("1"+chr(index(col(),+6,row()+38)))
```

This produced the required result, but it will not all print out if F for formulae is selected - only the first 60 characters will print.

Another spreadsheet I use is a monthly account, and I wished to add each month's totals to an annual summary. Again the number of the month in cell G1 produces the correct heading on the monthly accounts. The coding is best explained with an example. Row 41 is the annual summary for a particular month; the name of the month is in cell B41 and one of the End of Month totals is to be taken from the monthly account helf in cell D9 and copied into D41, but only if it is for the correct month. I coded D41: 'if(B41=month(G1),D9,D41)'. The result is a little disconcerting. On the screen the cell D41 may contain, say, £45.00, but if the cursor is put on D41, the 'contents' at the bottom of the screen contain the coding above. The only way to zero out the annual summary at the start of a new year is to zero-fill all monthly totals, then cycle cell G1 through from 1 to 12.

E E Stocker
Waterlooville
Hampshire

Trumpeted

May I, through your column, express my sincere thanks to Steven Honeyball of Miracle Systems who resolved a Trump Card disk drive interface problem for me free of charge. The fact that I had purchased it second hand and that it was brought to the latest Mod state makes it all the more impressive and a refreshing change in an age of indifferent service.

Keith Lock
Portwell
Dorset

Comment: Yes, gladly. This is actually the second letter I have had recently thanking Steven for his help and assistance. Which means that ... either Stuart has opted for a name-change after all that publicity we gave him back in April ... or that we have finally discovered how he gets through enough work for two men!

Save and

When you write a program you should save it often, especially when you test and debug it. Sometimes it may be hard even to remember which program you are working with. I have made a little procedure to do the deleting and saving job.

```
10 DEFine PROCedure sav(name$)
20 IF LEN(name$)<1 THEN
60 DELETE "device"&"myprog"
70 SAVE "device"&"myprog"
50 ELSE
30 DELETE "device"&name$
40 SAVE "device"&name$
80 END IF
90 END DEFine sav
```

Line 20 checks that if there is no given name together with procedure name example sav " ", then is first deletes the old "myprog" program and saves a new version of it. Else, if you have typed sav "test1" with a name in it, it deletes the program named "test1" and saves the new version of it. This way, you can save different versions of the program easily. When I start to write the program I first load that procedure and chance the name "myprog" to the name of my working program. I haven't seen such a program yet in *QL World*, so if you are interested to print it, it has been a great help to me.

Kalevi Hämäläinen
Tampere
Finland

Editor's comment: Backing up files in development under Qdos has always posed a puzzle. The standard solution is to save the file under alternate names, deleting the last-but-one file as you go. Using the routine above saves time but has two risks: should the power fail, or the computer lock, while the delete-and-save is in progress, both files will be lost. Also, if a test file with the same name exists on your directory (for example, from another program you have been developing recently) that file will be overwritten without warning.

What is needed is a similar routine which will re-name the previous file as a backup, and write the current file to the main filename, deleting the previous backup file (last file but one), reducing the danger of losing all files if a malfunction occurs during the saving process. Checking for duplicated file-names during a delete-and-save process is another story.

Big words

There seems to be an implicit belief (for instance, Bryan Davies mentions it in the February *QL World*) that the larger the dictionary in a spellchecker, the better. Digital Precision's *Perfection*, and the new version of *text87* offer vast dictionaries, but for most users these are actually less useful than smaller ones.

There is a difference in purposes between a paper dictionary and an electronic one. The first provides us with words we do not know; the other traps wrong spellings of words we do know. Some of those incorrect spellings will form real words, maybe unknown to us, but to be found in a large dictionary, and so passed as correct by a spellchecker using such a dictionary.

For example, if I were to type 'dod' instead of 'did' - an easy slip a small spellchecker would throw this up, but a large one might not, becasue such a word exists. My 12-volume *OED* has a third of a page giving seven definitions of 'dod'.

There is a need to study the optimum size for a spellchecker. Clearly it should include all the words we know, and none that we do not, but since we all have different-sized vocabularies perfection (with a small p) is impossible. I believe the best compromise is a dictionary of about 15,000 words with the facility for the user to add his or her specialities.

Ken Whyld
Caistor
Lincoln

Editor's comment: Perfection (with a big P) offers a choice of spellchecker sizes from moderate to very large: the new text87 Plus4 offers its English spellchecker in a choice of two sizes. The ability to add to (or edit) the dictionary is personal requirements is perhaps the most important facility in any spellchecker.

Seconds out

With reference to the article *Three in the Hand* by Mike Edwards in the April *QL World*, I was surprised to read Mr Edwards' comparative speed trials of similar functions on the three wordprocessors in question. I have no experience of *The Editor* or *text87*, but have used *Perfection* a great deal and am keen to defend it.

Mr Edwards claims that to move a block of text or to relate a string takes Perfection 25 seconds. From my experience that this is untrue. Even while working on long files, I have found that Perfection can move a block of text, practically independently of its size, position and distance moved, in under three seconds. I have also found Perfection to be equally fast in replacing strings, although this is obviously more dependant on the particular variables involved. Perhaps Mr Edwards forgot the decimal point in some of his results!

The article also suggets that it is impossible to change the printer driver data during each printing session. To put the record straight, different printer data files can be brought into use at any time through the normal menu system.

John Bunce
Mill Hill
N London

SOFTWARE FILE

POLYTEXT

Rich Mellor finally gets his very own column! (This month only.)

INFORMATION
Program: Polytext
Price: £17.50
Supplier: CGH Services, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA Tel (0559) 384574.

Many QL users have, over the years, become accustomed to *Quill's* very own way of working, and tend to be loathed to change, despite the idiosyncrasies of this now somewhat dated program. One of the main features missing from Quill is the ability to present text in column formats (such as you would find in a newspaper or *QL World*).

Reprocess

Nick Ward has produced *Polytext* with such users in mind. It cannot alter Quill itself to produce columnar output; instead it takes the Quill generated '_lis' files and then pre-processes them prior to sending them to a printer to give the desired number of columns. What is more, Polytext has a limited ability to allow you to merge small pictures with the text, thus enabling the user to create output which is both varied and a little more professional in appearance than would otherwise be the case.

Polytext is supplied as a collection of various files on a disk (or microdrive), together with a 14-page A5 manual. The manual would appear at first glance to be more than adequate, but, unfortunately, if you do experience problems when using the program, the manual is of little help (the problems mainly seem to relate to failings of the program rather than of the manual), and recourse to the publishers is the only real answer.

A boot file is supplied on the disk version of the program (a minor contribution by yours truly), which enables you to access easily Quill and the various utility programs provided by Polytext. Although each of the Polytext programs can be multitasked, Quill itself will get in the way unless it has been patched to stop it from grabbing all the memory.

You are better off setting up a separate disk for use with Polytext, containing the different Polytext files and the Quill program file. This is because the standard Quill program will only recognise 'printer_dat' as its printer driver information file, and any files which are to be processed by Polytext must use the special version of this (which is supplied).

The manual explains that any documents written within Quill must be altered slightly if they are to be processed by Polytext. The amendments necessary are very minor. Once you have written your document in Quill as normal you will then need to plan how your document is to be laid out (ie which parts are to appear in columns and how many columns). This should really be done by hand on paper so that you can decide the optimum column widths and gaps between columns.

Amend

Once the layout has been planned, you will then need to amend your Quill document (from within Quill) by typing in the various Polytext commands. The commands appear in curly brackets and are only necessary where the layout is to change. These commands tell Polytext whether to justify right, centre or left, the number of columns to print with the width of the columns and the margins, the typeface and the name of a graphics image to merge with the document.

The justification commands (eg {~PJR}) are quite simple to use, but as with all Polytext commands (other than the graphics command), they must appear on a separate line above the pharagaph(s) to which they relate.

The most complex of the other two sets of commands is the command which sets up the format for columnar output. This takes five parameters specifying the number of columns, the gutter, the column width, the number of lines between changes in layout, and the text type. Polytext can handle up to seven columns across a page, but on a standard 80-column printer three or four columns will really be the acceptable limit. The size of the gutter is up to you, but there is no option to set a larger left margin than a right margin and so whatever figure you decide upon, your text will appear centred. The formula needed to calculate the maximum width of a column is actually different to that printed in the manual and is in fact:

It is quite simple to go into various columns of printing. Although you cannot see anything in Quill as you type the document (this would require a complete re-write of the Quill program), you can put a Polytext command at the start of each paragraph, telling Polytext how to format the text once it has all been typed and is ready for printing out. All of Quill's normal features are available as if you were using Quill's very own printer_dat file. These of course cover high script, underline, Bold, and low script, as well as all of the possible combinations.

On top of this Polytext allows you to format the different paragraphs in Normal plus Elite sized print, compressed print, double height and double width (although of course, this will depend very much on your printer's own abilities).

POLYTEXT DEMONSTRATION

The title was produced quite simply by using the normal Bold and Underline commands of Quill, but also including two Polytext commands to ensure that the title was in the centre of the page and was in enlarged text.

((column width+1)*number of columns)-1+(2*gutter width)< line length

For those users who find it difficult to calculate the column width, one of the programs in the Polytext package, reck_obj, will work out the optimum size for you. This is a very simple program and although the manual does not go into depth about its operation, it is easy to use.

Last, but not least, the user has to include the 'text type' in the Polytext command. This tells Polytext whether to print the next few columns in one of five text styles (normal, elite, condensed (or reduced), double width or double height); although the actual effect of these styles will depend upon your printer (some printers, for example, can print double-height, single-width text, whereas others can only print double-height, double-width text). Unfortunately this text type can only apply to the whole of a paragraph (although Polytext will support the normal Quill typefaces within a paragraph as well as NLQ and Italics).

The figures show prepared for Polytext output. There are some odd characters at the top and end due to certain of Polytext's idiosyncrasies (see below).

Print

Adding pictures to a Polytext document is also quite easy – you merely need to include in your document a command similar to {PG:f1p1–example–pic). Some users may find that the printer driver falls down on their machines in this area, and it cannot be altered by the user (unlike the text printer driver). However, if you have an Epson compatible printer, you should not have any difficulty (the program worked fine on my MT-81). The program could however do with a little more development in this area, since the graphics printer driver is quite slow and can only handle simple pictures – and is suitable for logos or simple clipart. The program does not handle colour well and the user will tend to be restricted to black and white images.

Text types

By way of assistance, a simple program is included in the package to 'grab' parts of QL screens for use by Polytext. This program is again relatively simple to use – you merely have to load the program and enter the name of a screen file. You can then choose the part of the screen to be 'grabbed' by placing two crosses at two diagonals of the required area. I would have preferred this to have been achieved using a rubber-banded box, but then this package is never quite as easy to use as it should be. The actual area allowed is disappointingly small, but should prove adequate for relatively simple pictures when used in a multi-column part of a document.

Anyway, having prepared your Quill document, you will then need to Print the document to a file, using the special version of PRINTER_DAT supplied with the package. Having done this, you can leave Quill and enter the main Polytext program.

The main Polytext program starts up with a small window which invites you to set certain parameters. You will need to enter the name of the document (the '_lis' file produced by Quill) and check that the page length, line length and printer control filename are correct. Polytext does not have any defaults for the filename and therefore filenames should be entered in full (including the drive and _lis suffix). You can also set the top and bottom margins, the start page number and the footer (no headers are supported). The footer will need to be in the form of text followed by the page number, followed by more text if required, and can be left, right or centred. This allows as much complexity as Quill, but nowhere near as many options as *Perfection* (by Digital Precision), for example.

The program can then be run (press R), and it works quite slowly, reading in the document paragraph by paragraph and reformatting it as required. The program shows the document on screen as it is processed (although this is definitely not WYSIWYG) so that you can get some idea of whether your format commands are having the desired effect. This generally works okay, but I was somewhat surprised to find that the program failed with an error ('Index out of Range') on an extra paragraph which appeared in the original of Figure one (this was without apparent reason and deleting the paragraph allowed the program to continue without any further problems).

At the end of each page, the program gives you the option to print out the page, see a 'cameo' of the page or go onto the next page. Requesting a

Cameo

cameo will give you an idea of how the whole page will be laid out (although it might have been better to leave this on display while the page was being processed) and you can then decide whether it will look too cluttered before deciding to print it out. **Figure two** shows a page of the final formatted document.

Users who have a slightly different printer to those which the program currently supports may be able to get a printer control file from the suppliers, but for the unfortunate few who are still having problems, you will need to alter the file 'PRINT_DAT' supplied with the disk (this should not be confused with the Quill PRINTER DAT file which should not be altered). The manual explains that to alter the printer control file you will need to load it into a text editor or import it into Quill.

The actual control file consists of various descriptions followed by printer command sequences. The sequences should be altered to suit the codes listed in your printer manual (if you are having difficulty, I am certain that the suppliers will help you), and control such things as the end-of-line commands, commands to switch on and off bold, underline, near letter quality, italics, condensed, double-height, double-width, elite, subscript and superscript. You can also alter the commands needed to switch your printer into graphics mode (assuming it supports a 480 pixel per line graphics mode). Unfortunately, the control file (and Polytext) does not support translates (you therefore cannot generally print both £ and # in the same document), preamble or postamble codes. This is a major omission especially in view of the fact that the status of the printer when Polytext starts its work will inevitably be a moving target. The simplest answer to the lack of a preamble/postamble code is to include them in your Quill documents – looking in the QL manual (concepts section) you will see that you need to press CTRL ; to get ESC. You can therefore build up the standard printer initiate sequence by typing CTRL ; and then @ (giving ESC,@).

Polytext does, however, use four translate sequences to switch on and off italics and NLQ. It is therefore possible to turn these on and off from within a paragraph, but it is somewhat obscure that the user cannot supply their own translate sequences, especially since Quill would normally support 10 such sequences!

Dependent

Overall, Polytext is a nice idea but could do with certain extra features to make it really useful. The actual output produced is highly dependent upon the type of printer being used and therefore only those users with the more recent printers will be able to make the most of the program. It often strikes me as a set of routines looking for a program.

So what is my conclusion ?

Polytext takes Quill a long way from what it once was. It would have been nice if the actual columns were visible from within Quill as you were typing, but the method Polytext uses to get around this should not pose

Just for Starters

In the third and last part of his Basic programming tutorial, Don Smith adds two remaining procedures.

This final article outlines the missing procedures: alter and abstract. There are various methods for altering the fields of records, some quite elaborate with either a moving asterisk or cursor with special windows for effect. A simple and perfectly adequate one numbers each field, and only requires a single keypress to initiate alteration. I suppose my programming is somewhat spartan, but then I am concerned with results rather than artistry.

Data Jumping

Line 2090 indicates that you can jump through the data in large or small steps. Pressing Tab moves forward 625 entries in one go, and F1 back the same distance. The remaining function and cursor keys reduce the jumps to a final one step at a time. SELect operates a second time to prevent overstepping the boundaries. I have deliberately omitted an insert, since it is almost as quick to use the menu's 'Enter data' to add the extras and then sort. You can also use the Alter procedure as a viewfinder, without making alterations, simply to nip about your entries, checking here and there. This is quicker than screen list. Of course, screen list could be improved by inserting a line at 825 to enable listing to commence from any chosen point and then emending lines 830 and 850.

```
825 input#0,'From which line?';to 18;s: cls#0
830 for x=s to cost(0)
850 if(x-s+1)/20=int((x-s+1)/20) then
```

By now you may begin to appreciate how addictive programming can be, like an art form, always capable of improvement. A useful abstract procedure is much more difficult to plan for the reader because of the wide variety of choices. I have chosen as an example to pick out the titles by a given author, but it could equally have been books of a certain vintage, authors covering certain topics, etc. The bulk of the procedure would still remain the same, only those lines where a comparison is made would need changing.

There are two ways to abstract. You can start from the beginning, finishing at the end. The time taken is quite bearable even for a few thousand and, more importantly, the list can be unsorted. The second method requires a sorted list, where a binary search locates a required entry in a fraction of a second. The program then backtracks to the earliest correct entry, then forward to the last one. I have given you the option of either and also the option of printing to screen or printer.

Binary Quirk

The binary search has a built-in quirk. If you miss out the 0.5 in line 2610 and your required entry happens to be the very last entry, it will never find it. Similarly, with the 0.5 in place, the first entry will never be found, which is the reason for line 2600.

Suppose that you had 10 items and the

Listing One

```
2020 :
2030 rem------ALTER DATA----------
2040 defproc alter
2050 x=1
2060 rep loop1
2070  alterline
2080  rep loop2
2090   cls#0: print#0,'TAB,F1-5,cursors - 625,125,25,5,1,
       menu - ESC, alter - press number, delete - D':
       i=code(inkey$(-1))
2100   sel on i
2110    =9: x=x+625: =232: x=x-625: =236: x=x+125
2120    =240: x=x-125: =244: x=x+25: =248: x=x-25
2130    =200: x=x+5: =192: x=x-5: =216: x=x+1
2140    =208: x=x-1: =27: exit loop1: =49 to 52,100: exit loop2
2150   end sel
2160   sel on i
2170    =9,236,244,200,216: if x>cost(0): x=cost(0)
2180    =232,240,248,192,208: if x<1: x=1
2190   end sel
2200   cls: alterline
2210  end rep loop2
2220  cls#0
2230  sel on i
2240   =49: print#0,aut$(x)\fill$('-',25): input#0,aut$(x)
2250   =50: print#0,title$(x) fill$('-',40): input#0,title$(x)
2260   =51: print#0,year%(x) fill$('-',4): input#0,year%(x)
2270   =52: print#0,cost(x) : input#0,cost(x)
2280   =100: less=cost(0)
2290   for w=x to less
2300    at 18,40: print w: aut$(w)=aut$(w+1): title$(w)=
        title$(w+1): year%(w)=year%(w+1): cost(w)=cost(w+1)
2310   end for w
2320   cost(0)=less
2330  end sel
2340 end rep loop1
2350 cls#0: menu
2360 enddef
2370:
2380 rem------LINE FOR ALTER----------
2390 defproc alterline
2400 cls: at 6,10: print'Entry  ';x to 30;'1   ';aut$(x)\\to
     30;'2   ';title$(x)
2410 print\to 30;'3   ';year%(x)\\to 30;'4   ';cost(x)
2420 enddef
```

Listing Two

```
825  input#0,'From which line?';to 18;s: cls#0
830  for x=s to cost(0)
850  if(x-s+1)/20=int((x-s+1)/20) then
```

required entry was the 10th. The search starts off with low=1 and high=10, so the variable 'av' commences as (1+10) divided by 2=5.5 with 'int' removing any decimal places, making it 5. Line 2630 lifts 'low' up to value 5. We now have (5+10+ divided by 2, then 'int', giving 7. If you continue, you will find that you cannot get beyond 9! The binary search operates within a REPeat loop and only needs around 20 loops to find an item in half a million, all in a fraction of a second. This is why line 2640 counts each looping, because if you haven't found it by the time 'count' equals 20, then it isn't there. Line 2660 checks this and if a value greater than 20 is noted the program switches via the alternative at line 2930 and prints a message to screen. If an item has been found that corresponds to the search, then from line 2670 a search is made backwards to find the first mention. Having located it, at 2740 searching proceeds forward to the final appearance of the required name. Now knowing 'start' and 'finish' line 2810 proceeds to print out data.

Printer Tips

One very useful tip if producing much printer hardcopy is to be found in line 3260. The Epson handbook, which was very easy to follow, gave clear instructions for setting tabs horizontally across the page, using chr$(9). I found however, that whenever printed data came close to the next tab setting, the printer invariably overshot the mark and then backtracked to the correct tab setting. This meant that instead of a smooth passage of the ribbon cartridge from one end to the other, it proceeded in a series of jerks, back-forth-back-forth, taking twice as long to print. If you consider the second fill$, which prints blank spaces as they all do, I have allowed 27 spaces for author, which itself has just 25 maximum characters. By doing a subtraction, 27 minus the length of the author, the result is how many blank spaces before the next item so that all columns are in line. Also it gives a quick check of how much space the printout will take. In line 3260, we have 5+27+42+6+ cost, which equals 80 plus characters. Since Elite (line 2570) allows 97 characters per line, we have plenty of space.

If you are a novice reader and you have successfully got the program up and running, sufficient techniques have been covered to enable you to string together other programs for your own use. Moreover, when you write your own programs, nothing can go wrong that you can't go back and correct - and improve. You are in control. The QL even usually tells you which line it cannot accept so that you can list or edit the line to find out your mistake. In the final analysis, one's whole thought processes are subtly altered (for the better), the more programming you undertake - an alternative to yoga!

Listing Three

```
2430 :
2440 rem------ABSTRACT MENU----------
2450 defproc abstract
2460 at 4,35: under 1: print'SORTED LIST': under 0: print\\
     to 15;'A'to 20;'Author'
2470 at 9,35: under 1: print'RANDOM LIST': under 0: print\\
     to 15;'B'to 20;'Author'
2480 j=code(inkey$(-1))
2490 sel on j
2500  =97: sortedaut: =98: randomaut
2510 end sel
2520 enddef
2530:
2540 rem------ABSTRACT SORTED AUTHORS----------
2550 defproc sortedaut
2560 cls: cls#0: input#0,"Author's name(with or without initials"
     to 45;nam$
2570 print#0,'To screen - 1, to printer - 0': screen=inkey$(-1):
     if not screen: elite: space12
2580 high=cost(0): low=1: count=0: long=len(nam$)
2590 rep binary
2600  if aut$(1,1 to long)=nam$: av=1: exit binary
2610  av=int((low+high)/2+.5)
2620  if nam$=aut$(av,1 to long): exit binary
2630  if nam$>aut$(av,1 to long): low=av: else: high=av
2640  count=count+1: if count>20: exit binary
2650 end rep binary
2660 if count<=20 then
2670  rep back
2680   if av>1 then
2690    if aut$(av,1 to long)=nam$: av=av-1: else: start=av:
        exit back
2700   else
2710    start=av: exit back
2720   endif
2730  endrep back
2740  rep forward
2750   if av<cost(0) then
2760    if aut$(av,1 to long)=nam$: av=av+1: else:finish=av:
        exit forward
2770   else
2780    finish=av: exit forward
2790   endif
2800  end rep forward
2810  for x=start to finish
2820   if screen then
2830    seeline
2840    if (x-start+1)/20=int((x-start+1)/20) then
2850     cls#0: print#0,'Press any key to continue or ESC -
         back to menu'
2860     c=code(inkey$(-1)): if c=27: menu: else: cls
2870    endif
2880   else
2890    prinline
2900   endif
2910  endfor x
2920  if screen: cls#0: print#0,'Press any key for MENU':
      pause: menu: else: formfeed: menu
2930 else
2940  cls: for y=1 to 3: at 15,30: paper 7: ink 0: print
      'NAME NOT FOUND': pause 50: cls: pause 25
2950 endif
2960 paper 2: ink 7: menu
2970 enddef
2980 :
2990 rem------ABSTRACT UNSORTED AUTHORS----------
3000 defproc randomaut
3010 cls: cls#0: input#0,"Author's name, with or without initials"
     to 45;nam$
3020 print#0,'To screen - 1, to printer - 0': screen=inkey$(-1):
     if not screen: elite: space12
3030 long=len(nam$): found=0: search=0
3040 for x=1 to cost(0)
3050  if aut$(x,1 to long)=nam$ then
3060   if screen then
3070    seeline: found=found+1: search=search+1
3080    if found and found/20=int(found/20) then
3090     cls#0: print#0,'Press any key to continue or ESC -
         back to menu'
3100     c=code(inkey$(-1))
3110     if c=27: menu: else: found=0: cls
3120    endif
3130   else
3140    prinline
3150   endif
3160  endif
3170 endfor x
3180 if search=0 then
3190  cls: for y=1 to 3: at 15,30: paper 7: ink 0: print
      'NAME NOT FOUND': pause 50: cls: pause 25
3200  paper 2: ink 7
3210 endif
3220 if screen: cls#0: print#0,'Press any key for MENU': pause:
     menu: else: formfeed: menu
3230 enddef
3240:
3250 defproc prinline
3260 print#4,x;fill$(' ',5-len(x));aut$(x); fill$(' ',27-len(aut$(x)));
     title$(x);fill$(' ',42-len(title$(x)));year%(x);
     fill$(' ',6-len(year%(x)));cost(x)
3270 enddef
```

Abacus: only following orders

I P H Warne turns his Abacus spreadsheets to the task of keeping track of standing orders and other money movements.

Nigel Bates' article in the July 1991 *QL World*, on using *Abacus* to reconcile his bank statement, was a useful example of the simplest use of a spreadsheet. But most of the hard work was still being done on the stubs of his chequebook. One direct debit left out of the reckoning can give you a nasty shock when you find yourself in the red!

The spreadsheet described here is a means of keeping a regular check on your bank account to avoid that situation. Personally, I need to know in advance if I have to transfer money from my savings to my current account to avoid bank charges. Practically all our living expenses, apart from food, are settled by standing orders or direct debits, including Poll Tax, water rates, electricity, gas, car purchase, insurance, mortgage, etc. When trying to estimate what the state of the account will be on a certain date, it is easy to forget one or another of these payments. Others may wish to know how much they can safely withdraw from their current account for savings and investments, or to buy an expensive item.

The finished product will not look like a bank statement, with all entries in chronological sequence and three columns from Credits, Debits and Balance. Instead, this spreadsheet has two columns for each month (one for text, with credits and debits in two blocks. The sheet also overlaps into January of the following year to allow for checking until the bank sends the final statement for the year.

The biggest problem concerns regular payments and/or credits, particularly those due quarterly or biannually. For an account spreadsheets to be useful, all thee should be entered automatically once the due date in each month is passed. So it may be helpful to analyse the conditions which must be met for a debit or credit to be included in the account column for a particular month:

1. The date of the account must be on or

TABLE 1

```
Column widths (use F3 G W nn col ENTER col ENTER):-
    A 16; C 1; E 1; all others remain at 10 unless you wish
    to increase the width of the monthly 'text' columns (H,
    J,L,N, etc.). A width of 12 will still allow printout in
    condensed font across 3 A4 pages.
Numbers to 2 decimal places (F3 U D ENTER ENTER)
Settings via Design command (F3 D)
    Auto-calculate OFF (A)
    Zero shown as blank (B)
    Set any other design parameters you require (e.g.
    lines per page, characters per line) at the same
    time, then ENTER
```

TABLE 2

```
A1    Optional text (Account number, etc.)
E1    Optional text ("For current Balance go to cell AExx")
A2    Text "ACCOUNT FOR YEAR"
B2    Text "1991" (or current year)
A3    Text "DATE:-" (Right justified with F3 J ENTER
          ENTER R A3 ENTER)
B3    askt("Enter date, with format DD/MM/YY    ")
I3    month((col()-7)/2)
Copy (F3 C) I3 to K3; I3:K3 to M3; I3:O3 to Q3; I3:Q3 to Y3
Calculate (F3 X)
AF2   Text "YEAR" (Right justified)
AG2   str(val(B2)+1,2,0)
AH1   Text "DATE" (Right justified)
AI1   Text "CONSTANTS:-"
AI3   (N.B. see text for instructions on entering this)
      "19"+B3(len(B3)-1 to )+"/"+str(val(B3(instr(B3,"/")+1
       to ))+100,2,0)(2 to )+"/"+str(val(B3)+100,2,0)(2 to )
AI4   days(AI3)
AI5   val(AI3(6 to ))
AI6   val(AI3(9 to ))
AI7   int((AI5/3-int(AI5/3))*3+0.1)
AI8   int((AI5/6-int(AI5/6))*6+0.1)
Convert to integer display with F3 U ENTER I ENTER AI4:AI8
          ENTER
E4    Text "Balance Brought Forward:"
H4    Text "------"
I4    Balance
A5    Text "CREDITS"
I15   Text "--------" (Right justified)
G16   Text "TOTAL CREDITS:-"
I16   if((col()-7)/2>$AI5+if(val($AI3)=val($AG2),12,0),0,sum
      (I4:I15))
Copy I15:I16 to K15; I15:K16 to M15; I15:O16 to Q15; I15:Q16
       to Y15
AH16  sum(I16:AG16)
A17   Text "STANDING ORDERS & DIRECT DEBITS"
F17   Text "1st/Latest"
B18   Text "Start Date:"
D18   Text "Last Date:"
F18   Text "Payment:" (Centre justified)
G18   Text "Amount:" (Right justified)
A19   Text "1. MONTHLY:-"
B20   Date of 1st. payment in standard format
C20   if($AI4<days(B20),15,if($AI6<val(B20(9 to )),-1,0))
D20   Date of final payment
E20   if($AI4)=days(D20),15,if(C20+$AI5=0,12,0))
F20   if(C20=15,B20,if(E20=15,D20,str(val($AI3)+if($AI5=1,
      C20,0),2,0)+"/"+str($AI5+C20+E20+100,2,0)(2 to )+B20
      (8 to )))
G20   Amount of payment
I20   if(C20<15 and days($B2+"/01"+B20(8 to ))>=days(B20) and
      days($B2+"/01"+B20(8 to ))<=days(F20),G20,$H4)
similarly at K20,M20,O20, to AE20, except that for "/01",
      type "/02", "/03","/04", etc. N.B. DO NOT USE the COPY
      function. The formula in each column is different.
AG20  similar to I20, except that $AG2 is substituted for $B2
      in formula.
```

after the first payment is due;

2. The due date in that month must be on or after the date of the first payment;

3. The due date in that month must be on or before the current date;

4. The due date in that month must be on or before the date of the final payment.

These last two conditions are convergent, and may be restated as a single condition, viz:

3(a). The due date in that month must be on or before the date of the most recent payment.

The treatment of dates in Abacus is limited. Dates are ten character text strings with three possible display formats, but for use in calculations they must be in the format YYYY/MM/DD, with leading zeros if necessary. All other formats will be rejected, with error reports. The days() function will give the number of days elapsed since 1 January 1583, but there is no complementary function, as there is in SuperBasic, to give the date string from the number of days elapsed since that date. If one wishes to compare or manipulate years, months or days of the month, it is necessary to slice the date string and use the val() function, converting the result back to text, if required, with the str() function. Leading zeros are inserted by adding 100 to the modified val() and slicing off the first digit of the subsequent string. The disadvantage of all this is that it tends to make the formulae for date manipulation rather long and often longer than Abacus of the QL will accept (the Psion PC4 and Exchange programs will accept longer ones).

First and final

Two columns, B and D, are allocated to show the dates of the first and final payments of a direct debit or standing order. The restriction on length of formula has been overcome by having a 'flag' column for each of these dates, which a third date column uses to calculate the date of the most recent payment. The formula in each month's column to calculate compliance or non-compliance with the three conditions above then becomes relatively simple and brief.

The formulae have been drawn up to make date manipulation totally independent of a particular year. The dates shown in columns B and D can, therefore, be the true limiting dates for that particular payments; and to start a new year, it is only necessary to change the year in one call (B2). The only drawback to this is that the memory required is more than the 23K which a spreadsheet on an unexpanded QL is allowed. However, the formulae can be modified provided all the dates in columns B and D relate only to the current year and the thirteenth monthly column is omitted - more about that next month.

A small block in the top right hand corner of the spreadsheet shows the current date in the format Abacus requires, and below

```
Right justify text in cells I20:AG20
A21  Text "2. QUARTERLY:-"
B22:AG22 same as for B20:AG20 except for
C22  if($AI4<days(B22),15,int((val(B22(6 to ))/3-int(val(B22
     (6 to ))/3))*3+0.1+if(val(B22(6 to ))/3=int(val(B22(6
     to ))/3) and $AI5<=3,3,0)))
E22  if($AI4>=days(D22),15,if($AI5<C22 or ($AI5=C22 and $AI6
     <val(B22(9 to ))),-9,if((C22<>3 and C22)$AI7) or (C22=
     $AI7 and $AI6<val(B22(9 to ))),3,0)))
F22  if(C22=15,B22,if(E22=15,D22,str(val($AI3)-if(E22=-9,1,
     0),2,0)+"/"+str($AI5-$AI7+if(C22=3 and $AI5=3,0,C22)-
     E22+100,2,0)(2 to )+B22(8 to )))
and the formulae in cols.I to AG are only copied to the
     relevant months for that particular payment.
A23  Text "3. BI-ANNUAL:-
B24:AG24 same as for B22:AG22 except for
C24  if($AI4<days(B24),15,int((val(B24(6 to ))/6-int(val(B24
     (6 to ))/6))*6+0.1+if(val(B24(6 to ))/6=int(val(B24(6
      to ))/6) and $AI5<=6,6,0)))
E24  if($AI4>=days(D24),15,if($AI5<C24 or ($AI5=C24 and $AI6
     <val(B24(9 to ))),-6,if((C24<>6 and C24)$AI8) or (C24=
     $AI8 and $AI6<val(B24(9 to ))),6,0)))
F24  if(C24=15,B24,if(E24=15,D24,str(val($AI3)-if(E24=-6,1,
     0),2,0)+"/"+str($AI5-$AI8+if(C24=6 and $AI5=6,0,C24)-
     E24+100,2,0)(2 to )+B24(8 to )))
A25  Text "4. ANNUAL:-"
B26:AG26 same as for I22:AG22 except for
C26  if($AI4<days(B26),15,0)
E26  if($AI4>=days(D26),15,0)
F26  if(C26=15,B26,if(E26=15,D26,str(val($AI3)-if($AI4<days
     ($AI3( to 4)+B26(5 to )),1,0),2,0)+B26(5 to )))
A27  Text "OTHER DEBITS:-"
Copy I15:AG15 to I33
G34  Text "TOTAL DEBITS:-"
I34  if((col()-7)/2>$AI5+if(val($AI3)=val($AG2),12,0),0,sum
     (I18:I33))
Copy I34 to K34: I34:K34 to M34: I34:O34 to Q34: I34:Q34 to
     Y34
AH34 sum(I34:AG34)
Copy I15 to AH35
AF36 Text "BALANCE ON"
AG36 AI3(9 to )+AI3(5 to 8)+AI3( to 4)
AH36 AH16-AH34
AH37 Text "========" (Right justified)
and finally, convert figures in ranges C20:C26 and E20:E26
     to integer display format with:-
          (F3 U ENTER I ENTER C8:C26 ENTER).

Note:
     The formulae for weekly, fortnightly or 4-weekly debits
or credits are:-
Column F:
     if(val(Bxx)<val($B2),$B2+"/01/"+str(8-int(((days($B2+
     "/01/01")-days(Bxx))/n-int((days($B2+"/01/01")-days
     (Bxx))/n))*n+0.1)+100,2,0)(2 to ),Bxx)
Column I:
     if($AI4<days(Fxx),$H4,(int((if($AI4>days(Dxx),days(Dxx)
     ,$AI4)-days(Fxx))/n)+1)*Gxx)
 where xx is the row number and n=7,14 or 28 as appropriate.
```

that, the days corresponding to that date, the number of the month and the day of the month. These values are referred to by the formulae in the 'flag' columns and the column showing the date of the most recent payment.

For monthly payments, the 'flag' column C can have one of three values: 15 (a purely arbitrary value) if the first payment is not due until after the current date; -1 if the day of the month is less than the due date; or zero. The 'flag' in column E has a value of 15 if the current date is after the date of the final payment, 1 if the current month is January and the 'flag' in column C is 1, or zero. The formula in column F then selects one or other of the dates in columns B or D, if the value in either 'flag' column is 15, or else combines the values in them with elements of the current date to arrive at the date of the most recent payment.

By making the flag columns C and E one character wide and blanking zeros, either a hash (#) is displayed or nothing, in the monthly and annual payment sections. In the quarterly and biannual sections, a number up to 3, or 6, or a hash may show.

Define grid

To start off the spreadsheet after loading Abacus, use the details in **Table one** to define the grid and default settings. Then enter all the text and formulae in **Table two.** The dates you insert in columns B and D can be purely nominal, but there must be dates in them before you enter the relevant formulae or error messages will be generated. Type the formulae exactly as shown, taking care not to omit any brackets, commas, quotes or dollar-signs. Be careful not to confuse numeral 1 with capital letter I. There are a lot of references to cells in column AI (yes, AI) in the formulae, but none to bells in column A, which is only

```
 1|BANK ACCOUNT #12345678 - A.N.Y. BODY   For current Balance go to cell AE45
 2|ACCOUNT FOR YEAR 1991                                                                                    YEAR 1992
 3|         DATE|-   3/12/91                                                 January          February          nber          December          January
 4|                                          Balance Brought Forward|------      358.84
 5|STANDING CREDITS|-                          1st/Latest
 6|                  Start date|  Last date|    Credit|      Amount|
 7|MONTHLY|-
 8|Pension Fund      1991/01/16 # 1991/03/16 # 1991/03/16     466.37            466.37            466.37          -----             -----             -----
 9|                   1991/04/16 # 1992/03/16  1991/11/16     590.34            -----             -----           590.34            -----             -----
10|
11|OTHER CREDITS|-                                                                Cash 12/2        100.00 Cash
12|                                                                                                  Cash
13|
14|                                                                             -------           -------         -------           -------           -------
15|                                                  TOTAL CREDITS|-            825.21            566.37          590.34                                         £7686.65
16|STANDING ORDERS & DIRECT DEBITS|-            1st/Latest
17|                  Start date|  Last date|    Payment|      Amount|
18|1. MONTHLY|-
19|Ncle & G'd Water 1991/05/01   1991/05/01 # 1991/05/01     £23.82            -----             -----           -----             -----             -----
20|                  1991/06/01   1991/12/01 # 1991/12/01     £23.77            -----             -----           23.77             23.77             -----
21|Building Society 1990/10/03   1991/03/03 # 1991/03/03     £141.31           141.31            141.31          -----             -----             -----
22|                  1991/04/03   1991/07/03 # 1991/07/03     £126.43           -----             -----           -----             -----             -----
23|                  1991/08/03   1992/01/03   1991/12/03     £113.66           -----             -----           113.66            113.66            -----
24|Scottish Life     1989/04/03   1999/03/03   1991/12/03     £25.76            25.76             25.76           25.76             25.76             -----
25|H,F,C Trust      1989/07/14 # 1993/03/14   1991/11/14     £35.35            35.35             35.35           35.35             -----             -----
26|Poll Tax         1990/06/29 # 1991/02/29 # 1991/02/29     £31.20            31.20             31.20           -----             -----             -----
27|                  1991/05/29 # 1991/05/29 # 1991/05/29     £28.89            -----             -----           -----             -----             -----
28|                  1991/06/29 # 1992/02/29   1991/11/29     £28.50            -----             -----           28.50             -----             -----
29|2. QUARTERLY|-
30|Br. Gas Service  1991/02/11 2 1991/11/11 # 1991/11/11     £17.35                              17.35           17.35
31|3. BI-ANNUAL|-
32|Oxfam            1989/03/01 3 1993/03/01 6 1991/09/01     £40.00
33|4. ANNUAL|-
34|Union Subscrip'n 1990/01/31   1995/01/31   1991/01/31     £14.00            14.00                                                                             -----
35|National Trust   1990/09/28   1995/09/28   1991/09/28     £38.00
36|
37|CHEQUES (Payee and/or Number)|-                          Cash  5/1        150.00 BPOPB 6/2         7.64          120.00 Cash  2/12        100.00
38|                                                         BPOPB 8/1         15.84 Cash  6/2       200.00           23.56
39|                                                         B'card 370        34.30 B'card 371        45.00          100.00
40|                                                         Cash 23/1        250.00 B.S.S. 372         2.00
41|                                                                                Cash 16/2        20.00
42|                                                                             -------           -------         -------           -------           -------
43|                                                  TOTAL DEBITS|-             £697.76           £525.61         £487.95           £263.19                    £7493.46
44|                                                                                                                                                          ========
45|                                                                                                                                BALANCE ON 03/12/1991      £193.19
46|Figure 2 - specimen completed account.                                                                                                                    ========
```

used for text. Note that the entry in cell AI3 needs special attention because it begins with a quote. If the quote is typed first, Abacus will think that you are entering text, whereas you want to input a formula. The way round that is to miss out the initial quote until you have types in the rest of the formula; then go back to the beginning (with the UP-arrow) and type the missing quote. If desired, the amounts in the totals cells, and in column G, can be displayed in monetary format (F3 U M ENTER range ENTER).

Next month, we shall look at the formulae needed for payments at intervals other than monthly, and finish the spreadsheet ready for entering some real figures.

```
 1|ACCOUNT #12345678                  For current balance, go to cell AExx
 2|ACCOUNT FOR YEAR 1991                                                                                    YEAR 1992
 3|         DATE|- 26/9/91                                                   January          Februa          November          December          January
 4|                                          Balance Brought Forward|-----      200.00
 5|STANDING CREDITS                            1st/Latest
 6|                  Start Date|  Last Date|    Credit|      Amount|
 7|MONTHLY|-
 8|                  1990/01/01   1995/01/01   1991/09/01      50.00            50.00             50.             -----             -----             -----
 9|OTHER CREDITS|-
10|
11|
12|
13|
14|
15|                                                                             -------           -------         -------           -------           -------
16|                                                  TOTAL CREDITS|-            250.00            50.00                                                          650.00
17|STANDING ORDERS & DIRECT DEBITS             1st/Latest
18|                  Start Date|  Last Date|    Payment|      Amount|
19|1. MONTHLY|-
20|                  1990/01/01   1995/01/01   1991/09/01      50.00            50.00             50.00           -----             -----             -----
21|2. QUARTERLY|-
22|                  1990/02/01 2 1995/02/01 3 1991/08/01      50.00                              50.00           -----
23|3. BI-ANNUAL|-
24|                  1990/06/01   1995/12/01   1991/06/01      50.00                                                              -----
25|4. ANNUAL|-
26|                  1990/01/01   1995/01/01   1991/01/01      50.00            50.00                                                                             -----
27|OTHER DEBITS|-
28|
29|
30|
31|
32|
33|                                                                             -------           -------         -------           -------           -------
34|                                                  TOTAL DEBITS|-             50.00             50.00                                                          450.00
35|                                                                                                                                                          -------
36|                                                                                                                                BALANCE ON 26/09/1991      200.00
37|Figure 1 - the basic spreadsheet.                                                                                                                          =======
```

QL SCENE

UPDATE FROM DIGITAL PRECISION

Digital Precision has presented updated versions of a number of its programs, great and small. Perhaps of the most interest to many users will be the *Lightning Special Edition - Gold Card version*. The important feature of this version is that it costs less, as it does not require the rom normally supplied. The price is £39.95, or as a £25 upgrade from *standard* Lightning. Gold Card owners who already have the Special Edition rom can return the rom cartridge to Digital Precisionfor a £5 credit towards future orders.

Professional Publisher part two now has even more fonts ('the best high definition fonts yet') and utilities: extract screen from anywhere within the page; compress page; convert alien fonts. Part two costs £29.95 alone; £49.95 the pair.

Also for ProPub, the *Laserjet/ Deskjet/Inkjet driver* for HP printers gives superb output quality for £19.95.

QMaths part two contains optimally speeded Mandelbrot and Julia set visuals; enhanced statistical functions; terrain plotting; function evaluation, and more. Part two costs £59.95, or £99.95 with part one as well.

For DOS users, *Digital Research (DR) Dos 6.0* is now available, with many features advanced on DOS 5.00, including automatic disk compression on Write and decompression on Read, from 2x to 8x depending on the file, giving great storage space saving. There is also an on-line DOS reference guide. Many DOS users prefer DR-DOS to MS-DOS. Two manuals, a ready reference guide and a preconfigured ready-to-run QL/DR-DOS 6.0 disk is included in the price of £80, or £70 to listed customers who have purchased a previous DOS from Digital Precision. (DP notes that this is not a 'return-to-upgrade' arrangement - you keep your old DOS and get £5 off the new one as well.)

Certain combined prices are available if two packages are ordered at the same time: £179.97 for DR-DOS and *PC Conqueror Gold Special Edition*; £139.95 for DR-DOS and standard *Conqueror*; £109.95 for DR-DOS and *Solution*.

Orders and enquiries to **Digital Precision Ltd., 222 The Avenue, Chingford, London E4 9SE. Tel. 081 527 5493.**

Help needed with file conversion!

Would anyone with experience in converting QL-originated documents and listings for PC-compatible or Apple Mac systems please turn to the Editor's Notebook on page 14! - We are looking for advice.

Quanta workshop, Bristol

Henry Orlowski

One of the country's best Quanta workshops took place on 25/26 April at the Walton Park Hotel, a fine but unimposing building overlooking the sea in Clevedon, Avon. The meeting attracted a good bunch of enthusiasts from far and wide, despite relatively low-key publicity, a testimony to the reputation for quality and excellent organisation achieved by the Bristol QL User Group.

It's the prospect of meeting like-minded people, getting updates on the latest news and products, and the chance to spend some money that brings people to meetings like this.

Who was there? Digital Precision (with Freddy holding court), Miracle Systems, Dilwyn Jones Computing, CGH Services, Jochen Merz (all the way from Germany), EEC, TF Services and, of course, Quanta themselves. These 'big boys' were selling wares in good volume, especially the hardware suppliers like Miracle with their super Gold Card, and EEC with hard disks and interfaces. The Gold Card was reduced from £35 to a bargain £225 - worth having.

The 'small boys' were there too. Adman Services had all their usual goodies and free advice, and Ron Dunnett (Qubbesoft) with his new public domain disk including a ZX81 emulator! No stick-on 16K expansions, though, thankfully.

One of the simplest yet most effective gizmos I saw was the Rom Switcher sold by QLas. This switches between the Minerva rom and a JS (or JM), so that if you have programs that won't run under Minerva, and you don't want to incorporate the patch, you can switch back to your unenhanced rom. This is £25 from QLas.

No QL meeting would be complete without screwdrivers and soldering irons. Beginners were having problems sorted out by experts; there were on-the-spot repairs, ship changes and refits going on. There were QLs that didn't look like QLs, and four-foot-long QLs with miles of spaghetti cabling. These are the idiosyncrasies that make the machine the enthusiast's dream that it is.

When you've got all your upgrades and no longer need the old stuff, take it to the 'bring and buy' stall, where Mike and Mike will do their best to sell it for you. If you're starting on the first rung, this is the place to get your bargains: complete QL systems with expansion, disks, monitor and software for under £200, if you're lucky.

And finally, for those who like a different kind of excitement, Quanta held their AGM on the second day, and quite a rousing affair it was, too.

ARCHIVE

A=N+S-W=E+R-S

Robin Stevenson describes how to use your QL as a direct typewriter under full position-control from Archive.

This month's Archive Answer is a bit of an oddball as database applications go. In fact it isn't a database application at all. It is a mini application in its own right, more along word-processor lines.

One of the things a word processor is not too good at is fitting text onto a page which is already laid out. Often, it would be nice to use your printer to fill in an application form, or just add some more text to a page you have already printed. If you have ever tried doing this, guessing where the boxes or spaces would occur, you will know the chance of success is low. By comparison, an old typewriter could do the job tolerably well, because you can line up the paper and then type it in directly.

Well, we are not going to let a steam-powered lump of metal get one up on electronic wizardry. Anything it can do, the QL can do better, and here it is - the program to control your printer directly. You can position the paper and print head, type in text (with full editing facility); you can control underlining, emboldening, etc., and can even send printer escape codes if required.

Advantage

The advantage of using Archive instead of SuperBasic is the access to the Psior printer driver. Since every printer is different, and you have already customised

```
proc start
  let UNDER=0: let BOLD=0: let HIGH=0: let LOW=0
  let FLUSH$=" "+chr(0)+chr(8)    : rem SET PRINTER SPECIFIC VARIABLES
  let LINEUP$=chr(0)+chr(27)+"j"+chr(0)+chr(30)
  let BACKSPACE$=chr(0)+chr(8)
  let GOTOLEFT$=chr(0)+chr(13)        : rem PROVIDED AUTO-LF IS NOT ON
  Type                                 : rem LAUNCH THE MAIN PROCEDURE
  endproc
```

```
proc Heading;Text$
  if not UNDER: print chr(5);: endif
  print ink 6;chr(15);chr(27)+"C";chr(30);Text$; tab 67;chr(250);
  print " B"(BOLD+1);" U"(UNDER+1);" H"(HIGH+1);" L"(LOW+1);
  print chr(27)+"D"+chr(14); ink 5+BOLD;
  if not UNDER: print chr(5);: endif    : rem TURN UNDERLINE OFF AGAIN
  endproc
```

```
proc PrinTab;OldPos,NewPos
  if NewPos>OldPos                     : rem REPLACEMENT FOR LPRINT TAB
    lprint rept(" ",NewPos-OldPos);
    endif
  endproc
```

```
proc GetLine
  Heading;"Type the required text, and press enter."
  input TEXT$;                                : rem STANDARD TEXT INPUT
  endproc
```

```
proc AsciiCodes
  local Ascii$: let TEXT$="z": let Ascii$=""
  print chr(27)+"C"; at 0,0;     : rem STORE CURRENT SCREEN CURSOR POS
  while TEXT$<>""                : rem LOOP TO GET SEVERAL ASCII CODES
    print "Enter ASCII codes (end with blank) :"; tab 67;
    input at 0,40;TEXT$
    if val(TEXT$): let Ascii$=Ascii$+chr(0)+chr(val(TEXT$)): endif
    endwhile
  lprint Ascii$;: print chr(27)+"D"; : rem PRINT CODES, RETURN CURSOR
  endproc
```

```
proc LookUp
  local Key$     : rem SCROLLS PAPER UP FOR INSPECTION, AND RETURNS IT
  lprint : lprint : lprint : lprint : lprint
  Heading;"LOOK AT PRINT: press any key to continue"
  let Key$=getkey()
  lprint LINEUP$;LINEUP$;LINEUP$;LINEUP$;LINEUP$;
  endproc
```

```
proc TypeFace
  Heading;"TYPEFACE : [B]old  [U]nderline  [H]igh  [L]ow  on/off"
  local C$: let C$=upper(getkey())
  if C$="B": lprint chr(15);: let BOLD=BOLD=0: ink 5+BOLD: endif
  if C$="U": print chr(5);: lprint chr(16);: let UNDER=UNDER=0: endif
  if C$="L": let LOW=LOW=0: lprint chr(17);: endif
  if C$="H": let HIGH=HIGH=0: lprint chr(18);: endif
  endproc
```

your printer driver to suit yours, that is one of the major programming headaches deal with completely.

Well, almost completely. To give full paper control, it is necessary to add a few printer commands not covered by the printer driver. In particular, it uses backspace reverse paper feed, carriage to left; and printer buffer flush. In all these cases, you may have to consult your printer manual if you wish to use them.

The issue of buffer flushing in particular could give a problem. Some printers don't print every letter as it arrives, but store them in a memory buffer until a completed line has been received. If this is the case you must find a way of fooling the printer into sending these characters, so that the print head physically moves into place. One such way (as shown in the listing) is to send a space followed by a backspace. If that doesn't work you may need to hunt around the manual for an alterative.

Variables

All the printer-specific bits are entered as variable in proc Start. As listed, they should work with an Epson LQ emulation. Juki 6100 users should replace these with the alternative Start procedure shown in listing 2. Other printers may require something different again. Laser users are no-hopers on this particular application. The paper can not be seen or controlled in the same way. "Serves them right for having such a fancy kit," did I hear someone mutter?

The rest of the program is in listing one. There is no need to type in the remarks on the right hand side. These help explain what is going on, but fill up useful memory. Once you have antered, and saved it, you can try it out. All the key-strokes are echoed both on the screen, and on the printer as you type them. The four cursor keys should move the print head/paper as required; while the return key will send a new-line and return the print head to the left.

Tab Key

The tab key moves the head to the point you last entered some text, or if already beyond this, moves it a further eight places. Shift+Tab will home the cursor back to the left again and clear the tab setting. Shift+Down will give a FormFeed instruction. Shift+Up is the 'lookup' feature. It scrolls the paper round five lines, and waits for a key-press before scrolling it back again. This gives a handy way of checking you are typing in the right place, but should not be used near the bottom of a page.

The remaining functions are accessed by the function keys. F3 allows you to send Ascii codes to the printer. You enter each code number, in decimal, and finish with a blank entry. Only when you finish are the codes send, as a single string. These codes bypass the normal printer driver processing, so what you enter is what the printer will receive.

The F4 key gives you control over underling, bold, high and low script. As in Quill, you simply type the initial to turn it on and off. While a feature is on, its initial appears on the top right corner of the prompts line.

Finally, for all text entry, you press F5. You can then type in and edit some text. It will only be sent to the printer when you press return. Note that in this case pressing return does not give a new line, as you may wish to add more further along. A second return is needed, after you have finished entering text, to give a new line.

The system is designed for flexibility and control, rather than smooth efficiency. You could type a whole book using it, but I wouldn't recommend it. A few lines on a printed form are much more its natural home. One useful technique for accurate positioning is to enter a single full-stop as a test. Using the lookup feature you can then see if it is in the right place, and adjust the position accordingly before typing in the text itself.

Key Presses

If you are looking to extend the program yourself, one handy feature may be to send key-presses to a file, as well as to screen and printer. You could then use this to print out multiple copies, all correctly positioned on the form. With a bit of effort, you could even allow specific details to be typed at run time as well as standard text to be used every time. This would lift it onto another plane of functionality, but also of complexity. I have kept this listing short and simple, so it shouldn't be too daunting either to type in, or to understand.

```
proc Type
  local Pos,Left,H$,Control,Old: let Pos=0: let Left=0
  mode 0,8: print chr(20)+chr(4)+chr(2)+chr(77)+chr(23)
  let H$=chr(190)+chr(191)+chr(188)+chr(189)+" F5=text F4=type"
  let H$=H$+" F3=code SH +Dn=FF +Up=look +Tab=left ESC=quit"
  spooloff : let Control=60
  while Control<>27      : rem MAIN LOOP, FOR SUCCESSIVE INSTRUCTIONS
    let control=60:Heading;H$
    while Control>31: let Control=code(upper(getkey())): endwhile
    if Control=2: print chr(11);: lprint LINEUP$;: endif    : rem 2=UP
    if Control=3: print chr(10);: lprint :PrinTab;0,Pos: endif
    if Control=4: print " ";: lprint " ";PLUSH$;            : rem 3=DOWN
      let Pos=Pos+1: endif                                  : rem 4=LEFT
    if Control=5 and Pos: print chr(8);: lprint BACKSPACE$;PLUSH$;
      let Pos=Pos-1: endif                                  : rem 5=RIGHT
    if Control=30: print : lprint : let Pos=0: endif    : rem 30=ENTER
    if Control=29: let Old=Pos                          : rem   29=TAB
      if Pos<Left: let Pos=Left
        else : let Pos=Pos+8: endif
      print tab Pos;:PrinTab;Old,Pos: endif
    if Control=28: print chr(13);: lprint GOTOLEFT$;PLUSH$;
      let Pos=0: let Left=0: endif                          : rem  SH+TAB
    if Control=11: cls : print : lprint chr(12);: endif : rem SH+DOWN
    if Control=10:LookUp:PrinTab;0,Pos: endif           : rem   SH+UP
    if control=20:AsciiCodes: endif                     : rem 20 = F3
    if Control=21:TypeFace: endif                       : rem 21 = F4
    if Control=22: error GetLine: let Left=Pos: lprint TEXT$;FLUSH$;
      let Pos=Pos+len(TEXT$): endif                     : rem 22 = F5
    endwhile : mode 1,8
  endproc
```

```
proc start
  let UNDER=0: let BOLD=0: let HIGH=0: let LOW=0
  let FLUSH$=""                    : rem NO PRINT BUFFERING ON A JUKI
  let LINEUP$=chr(0)+chr(27)+"D"+chr(0)+chr(27)+"D"
  let BACKSPACE$=chr(0)+chr(8)
  let GOTOLEFT$=chr(0)+chr(27)+chr(0)+chr(9)+chr(0)+chr(1)
  Type                             : rem LAUNCH THE MAIN PROCEDURE
  endproc
```

SOFTWARE FILE

ROCKFALL

INFORMATION

Program: *Rock-Fall*
Publisher: CGH Services, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA
Price: £10 (Disk or mdv)
Memory: 128K

Rich Mellor gathers moss in a Boulder-Dash style game from CGH Services.

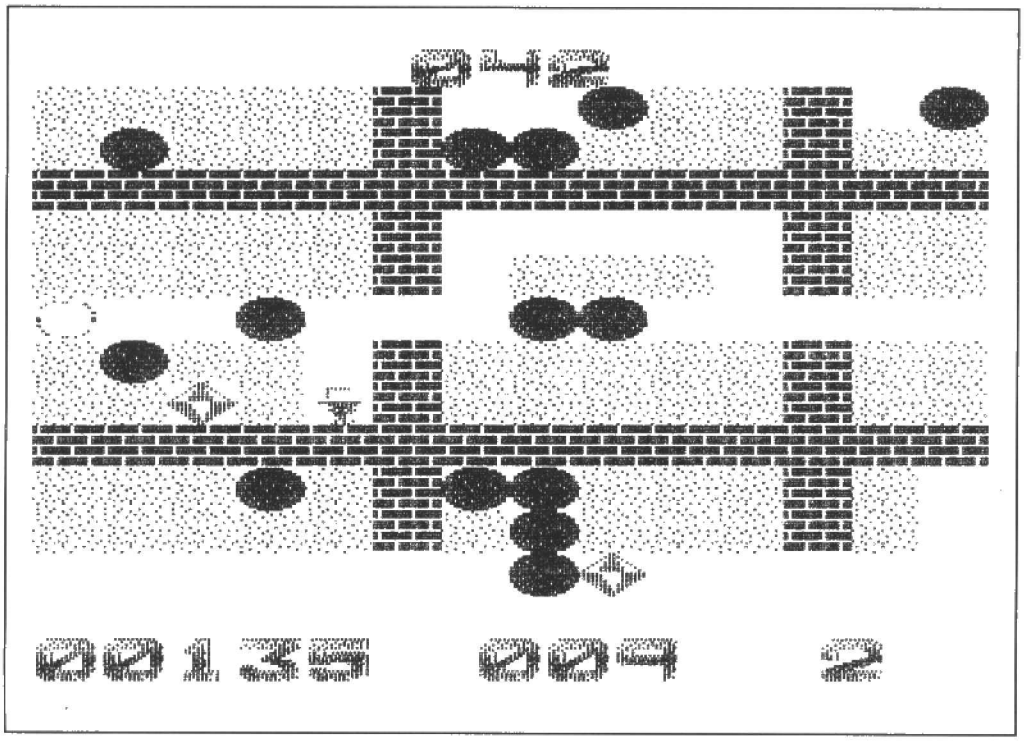
Many games players will recognise the theme behind this game if I say that it has been based upon *Boulderdash*. For those of you who have not heard of this game (which has almost become a genre in itself), you guide a little man around various screens collecting diamonds. The only problem is that between you and the diamonds is a load of earth which supports huge boulders. As you dig away at the earth (by moving your man across it), the boulders can become dislodged, causing a cascade of rocks and diamonds down onto your head and killing you, or blocking your path.

Rockfall is a very colourful variation on this theme and, together with falling rocks, you must also avoid two types of monsters who will chase you around the screen where the earth has been cleared away, and even walls which can grow across your only escape route. Monsters can be killed by dropping rocks on them, and thankfully, when they die, they will explode and (if you are lucky) take some of the walls with them.

Sixteen

There are a total of sixteen levels, although you can choose to begin your game on four of these, which will enable you to practise the more difficult sections. Each level is actually about four times the size of the QL screen, and as you move around more and more of the level is displayed. I am not however quite certain whether the lower levels are supposed to be easier, since I have only been able to complete two levels (level one and level nine - both of which are levels you can elect to start on).

In order to complete each level, you will need to collect a certain number of diamonds (the number of diamonds remaining to be collected is shown at the bottom of the screen). Luckily, you should not need to collect all of the diamonds on each screen and therefore early mistakes do not always ruin your chances of progressing to the next level. Also, in view of the fact that your attention will be more directed to the playing area rather than the bottom of the screen, there is an audible warning as well as a bar which is drawn across the top of the screen when you have collected enough diamonds. You may then proceed to the exit (which may not appear until enough diamonds have been collected) to enter the next level.

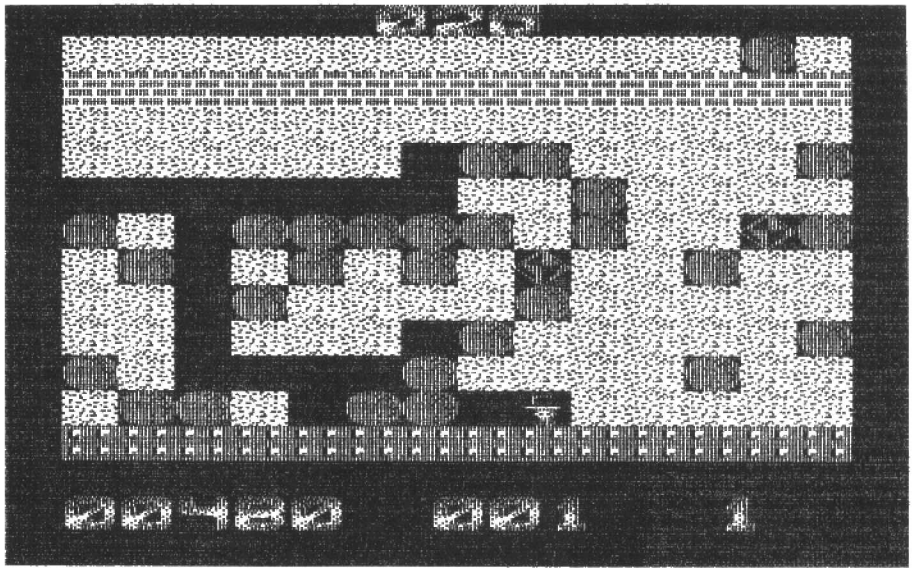


Amoeba

Some of the levels involve much more thought than others. For example, some of the levels incorporate what is described in the manual as a growing Amoeba. In order to complete these levels, you need to block the Amoeba's

growth by careful placement of the boulders. When it has finished growing, the Amoeba will change into the diamonds which you need to collect in order to complete the level. Now, this does require quite some thought when you need to stop an Amoeba that starts grows downwards, when all of the displaced boulders can only fall down!

Sound is quite limited in the game (no fancy background music here), and can be turned off by pressing the '\' key. You guide your little man around the screen with the cursor keys, and can even dig ahead of yourself by pressing the Space key at the same time as a cursor key. I would however have preferred to be able to re-define the playing keys so that I could play the game more easily on my full-sized keyboard.



Slug command

On a standard QL the game goes along at a medium pace, giving you plenty of time to think. However, Gold Card users may need to use quite a large value with the SLUG command (supplied on later Gold Card roms) in order to make the game playable. *(Do you remember when everyone complained that games were too slow?)*

The game comes with a small manual, which is adequate, but I believe that there will soon be incorporated into the program a screen giving an explanation of the symbols used by the game. As a whole, the game is well designed and plays very well. It is colourful and contains enough items of interest to keep younger players amused as well as older QL users. This certainly represents excellent value for money.

sinclair QL

STOP PRESS

All prices include 17.5% VAT

Falkenberg Hard Disk Drives

	Hard Disk Kit	Complete Set, boxed with PSU
32MB Hard Disk	£300	£365
20MB Hard Disk	£258	£322

The Hard Disk Kit comes without Box and PSU. If used with a gold card and bus expander the access time is greatly improved. (On a QL with Gold Card, approx 350,000 Kilobytes/second)

The QL Hard Disk Interface and Hard Disk controller boards can be mounted in various configurations as required. The following items are normally carried in stock and can be selected as required.

BUS Expander, 2 slots, (eg, for disk interface/gold card and hard disk controller)

Horizontal/ Vertical Connector £ 22.00 — Flexible cable Connector £ 30.00

BUFFERED BUS Expander, 5 slots, (For expansion other than Trump/Gold Card)

Horizontal/ Vertical Connector £ 59.00 — Flexible cable Connector £ 69.00

BUS Redirector, 1 slot, helps to reduce the width of extended QLs £ 17.00

RAM/ROM Battery Backed Memory Card (Ram to 896K, Eprom to 256K) From £ 79.00

For Option Details on any product send S.A.E.

Serial 3 Button Mouse £45

Remote Infra Red Mouse £55

Externally fitted into SER2, comes complete with interface and software. No Internal fitting required, 3 button mouse gives 5 more functions than the QIMI.

NB. We still have lots of QLs, 1 & 2 Mb disk drives, peripherals and components. See previous adverts or contact us for details.

EEC
18-21 MISBOURNE HOUSE
CHILTERN HILL
CHALFONT ST.PETER
SL9 9UE
TEL: 0753 888866

Payment terms:
CWO, Access, VISA et cetera
Cheques - allow 10 days
Delivery:
Carriage - £9 Postage - £4
Product subject to availability. E&OE.
FAX: 0753 887149

TF Services

MINERVA

The ULTIMATE operating system upgrade

MKII MINERVA rom with battery for 256 bytes ram, CRASHPROOF clock & interface (I²C) for robots etc. Can autoboot from battery ram.

Other features (common to MKI/MKII):
Autoboot or reset/ Multiple Basic/ Faster scheduler-graphics(10% of Lightning)-string handling etc/ WHEN ERROR/ Second screen/TRACE Foreign kybd drivers/ 'Worm' fast reset etc

MKI to MKI UPGRADES: up to 1.79 - £5. Rest FREE (send rom (PROTECT LEGS!) and disk or 2 mdvs)

Original MKI..... £40 MKII.....£65

ALL VERSIONS FULLY GOLD CARD COMPATIBLE

HERMES

A replacement QL co-processor (IC 8049 IPC)

Do you get keyboard bounce?
Do you find fast serial input unreliable?
Do you want to run an HSI 14400 bps modem?

If you can say YES to any one of these, then this is a product you have been waiting for

- Gives up to 19200 bps RELIABLE serial input
- Independent baud rates
- Stops keyboard 'bounce' (unwanted repeat chrs)
- Improves 'fuzzy' and 'random' sound
- Provides extra input and output lines
- Key click
- Reset/INT7 invoked more safely

FITTING is a simple job. Simply remove the QL top (8 screws) and replace the chip marked '8049' next to microdrive 1.

PRICE (including manual & software) £25

BULLETIN BOARD

GBBS - The first UK QL scrolling bulletin board. Now even longer with international ECHOmail and 10mb of files (including the UNCLE King James' bible!)

071-706 2379 V21/23/22/22bis 24hrs

QL SPARES

Faulty QL boards (excl plug in ICs).....	£9		
Keyboard membrane...	£8	Circuit diags...	£2
68008 cpu.....	£8	1377 PAL.....	£3
JM rom set.....	£10	Power supply...	£12
8302 ULA.....	£10	8301 ULA.....	£10
8049.....	£8	HDU ULA	£12

Other components (sockets etc)... plse PHONE

QuaLsoft

Viewdata/VT52/VT100 QL TERMINAL EMULATOR

Multitasking for electronic mail, PRESTEL etc, CET download. Compatible with Tandata Qconnect QL modem/modaptor. Use Qconnect RS232C to drive unbuffered modems. Comprehensive A5 manual.

Software (3.5" or mdv) & manual £30

FILE TRANSFER - to transfer text AND programs error free (XMODEM) between computers

Available for IBM compats, ATARI ST & Sinclair QL. Connects to Psion range with RS232C comms link. Software £7.50 per m/c. Cable £10 (name 2 m/cs)

QL REPAIRS

QLs tested with Thorn EMI test rig and ROM software

(MDU hardware extra if required)

£27 - 6 month guarantee

All prices include VAT and postage in the UK — MAIL ORDER ONLY

VISA

12 Bouverie Place, London W2 1RB (tel: 071-724 9853)

SCROLLING BBS FOR ORDERING/QL HELP and FAX - 071-706 2379

DIY TOOLKIT

Simon Goodwin looks through 'windows' for the QL and remedies some Qdos display problems he sees there.

Listing One

```
* Sinclair QL World DIY TOOLKIT - WINDOWS for QDOS
* Version 1.2, Copyright 1992 Simon N Goodwin
*
start       lea.l       define,a1       Point at names & addresses
            movea.w     $110.w,a2       Read the BP.INIT vector
            jmp         (a2)            Link SuperBasic extensions
*
* base_address = W_CRUNCH ( #ch% , flag% [0=All 2=Red 4=Green] )
*
crunch      bsr         get_channel     Pick up channel number
            subq.w      #2,d3           Check parameter count
            bne         bad_param2      Two are expected
            move.w      2(a1,a6.l),d6   Pick up compression flag
            addq.l      #2,$58(a6)      Tidy RI stack pointer
            bra.s       save_window
*
* base_address = W_STORE ( #ch% )       Store window contents
*
store       bsr         get_channel     Pick up channel number
            subq.w      #1,d3           Check parameter count
            bne         bad_param2      Only one is expected
            moveq       #0,d6           Flag for no compression
*
save_window bsr         find_window     Put details in registers
            move.w      d0,d7           Save width for later use
            moveq       #-1,d2          RAM owner is this task
            moveq       #24,d0          MT.ALCHP trap key
            trap        #1              Allocate Common Heap RAM
            tst.l       d0              Did that work?
            bne.s       exit            Return error code
*
            move.l      a0,d1           Preserve address of RAM
            move.w      d7,d0           Retrieve line word count
            move.l      #'Bufa',-4(a0)  Store flag for DISCARD
            move.l      224(a6),(a0)+   Point on to the next ALCHP
            move.l      d1,224(a6)      Extend the ALCHP list
*
            move.l      d5,(a0)+        Record total space used
            move.w      d4,(a0)+        Record line count
            move.w      d6,(a0)+        Record compression flag
*
* A4 -> First video word, A0 -> Buffer for screen information
* D4.W is count of pixel lines-1, D0.W is line width in words-1
*
            subq.w      #2,d6           Check the CRUNCH flag
            bmi.s       save_lines      If <2, store both bytes
            bne.s       crunch_line     >2, store green bytes only
            addq.l      #1,a4           Point at first red byte
crunch_line lea.l       save_bytes,a3   Point at byte saver
            bra.s       save_it
save_lines  lea.l       save_words,a3   Point at word saver
save_it     bsr         scan_lines      Call (A3) for each line
*
* RETURN_FP stacks D1.L in SuperBasic floating-point form
*
return_fp   move.w      d1,d4           D4 will be exponent
            move.l      d1,d5           D5 will be mantissa
*           beq.s       normal          A most unlikely case!
            move.w      #2079,d4        First guess at exponent
            add.l       d1,d1           Already normalised?
```

There has been much fuss in recent years about the rise of 'Windows' for computing applications; noisiest of all are the proponents of 'Microsoft Windows' for IBM micros and clones. That package bolsters the MS_DOS system with fashionable windows and icons, plus multi-tasking (programs and megabytes permitting) and device sharing - as long as the programmers resisted the temptation to write directly to the PC hardware.

It took them a while, but PC users are at least learning what QL owners have known for years - it is convenient and natural to run several programs at once, swapping between them at will. Microsoft Windows is slow and bulky, but that's great news for clonesters pushing their latest hardware expansion.

Alas, many good PC programs refuse to multi-task, because they were written for single-user systems and cannot share resources. Contrast this with QL tasks, which are concise and compatible, and it's easy to conclude that Qdos is still years ahead of the PC, especially if the 68008 is supplanted by one of the faster Motorola processors.

The QL has had multitasking and 'windows' from the start, unlike the PC, but the Qdos rom lacks facilities to save and restore the contents of displayed windows, so overlaps may disturb other displays.

The latest *DIY Toolkit* project remedies this lack, and adds a handful of new display routines to move, highlight, compress and re-colour QL windows. As ever, the code is concise, speedy and easy to use, on all Qdos systems; I have tested it on Minerva, Sinclair roms, and version 3.10 of the excellent Amiga Qdos emulator.

QL windowing systems are not new; they are components of *Qram*, *Qpac2*, and packages such as *Turbo Toolkit*. The DIY facilities build on their predecessors, packing twelve useful operations into just 676 bytes of machine code. They automatically pick up the window size and position from the channel definition, and work with Minerva's two screens - or even more, as I shall explain next month.

My code includes a neat facility to reclaim the memory used for stored screens when you load a new Basic program. Screens stored by conventional windowing packages can gobble over 32K of memory each, so the DIY keywords allow fast, reliable compression options that halve the space required. You can also manipulate stored windows with popular commands from Toolkit 2, Turbo Toolkit and DIY Toolkit. The Qdos RECOL command is flexible but rather sluggish, taking several seconds to recolour a large window. When many windows are in use it is convenient to have a quick way to highlight any particular one, so DIY Windows boasts a range of new re-colouring options, optimised for Mode4 displays

```
                bvs.s     normal
                subq.w    #1,d4              No, halve exponent weight
                move.l    d1,d5              Double mantissa to match
                moveq     #16,d0             Try a 16 bit shift
normalise       move.l    d5,d1              Take copy of mantissa
                asl.l     d0,d1              Shift mantissa D0 places
                bvs.s     too_far            Overflow: must shift less
                sub.w     d0,d4              Correct exponent for shift
                move.l    d1,d5              New mantissa is more normal
too_far         asr.w     #1,d0              Halve shift distance
                bne.s     normalise          Try shift of 8, 4, 2 and 1
*
* Check there's six bytes of space for the result
*
normal          moveq     #6,d1              Number of bytes needed
                movea.w   $11A.w,a0          BV.CHRIX vector
                jsr       (a0)
                movea.l   $58(a6),a1         Get a safe A1 value
                subq.l    #6,a1
                move.l    a1,$58(a6)         Grab 6 more bytes
                move.l    d5,2(a1,a6.l)      Stack mantissa
                move.w    d4,0(a1,a6.l)      Stack exponent
                moveq     #2,d4              Floating point result
                moveq     #0,d0
exit            rts
*
* GET_CHANNEL - evaluates integer channel number parameter
* Returns ID in A0 or returns error code to 4(A7)
*
get_channel     movea.w   $112.w,a2          CA.GTINT - get integers
                jsr       (a2)
                bne.s     quick_exit         Return D0 error code
                move.w    0(a1,a6.l),d7      Pick up the channel #
                bmi.s     bad_param          It should be positive
                addq.l    #2,$58(a6)         Update BV.RIP
default         mulu      #40,d7             Find offset in table
                add.l     48(a6),d7          Add base to offset
                cmp.l     52(a6),d7          Check not beyond end
                bge.s     chan_error         Number is too high
                move.l    0(a6,d7.l),d7      ID for output window
                bmi.s     chan_error         Channel is closed
chan_open       movea.l   d7,a0              Set up Channel ID
                rts
*
* W_SWAP #ch% - exchanges red & green display bytes
*
swap            bsr.s     get_channel        Read channel number
                subq.w    #1,d3              Check parameter count
                bne       bad_param2         Only one is expected
                bsr.s     find_window
                lea.l     swap_words,a3
*
scan_lines      lea.l     128(a4),a2         Point at the next line
                move.w    d0,d2              Duplicate width for loop
                jsr       (a3)               Call chosen line scanner
                movea.l   a2,a4
                dbra      d4,scan_lines      Process all the lines
                moveq     #0,d0              It worked!
                rts
*
* This listing will be concluded in my August DIY Toolkit column
```

Listing Two

```
100 REMark Sinclair QL World HEX LOADER v 3
110 REMark by Marcus Jeffery & Simon N Goodwin
120 :
150 CLS: RESTORE : READ space: start=RESPR(space)
160 PRINT "Loading Hex..." : HEX_LOAD start
170 INPUT "Save to file...";f$
180 SBYTES f$,start,byte : STOP
190 :
200 DEFine FuNction DECIMAL(x)
210 RETurn CODE(h$(x))-48-7*(h$(x)>"9")
220 END DEFine DECIMAL
230 :
240 DEFine PROCedure HEX_LOAD(start)
290 byte = 0 : checksum = 0
300 REPeat load_hex_digits
310    READ h$
320    IF h$="*" : EXIT load_hex_digits
330    IF LEN(h$) MOD 2
340       PRINT"Odd number of hex digits in: ";h$
350       STOP
360    END IF
370    FOR b = 1 TO LEN(h$) STEP 2
380       hb = DECIMAL(b) : lb = DECIMAL(b+1)
390       IF hb<0 OR hb>15 OR lb<0 OR lb>15
400          PRINT"Illegal hex digit in: ";h$ : STOP
420       END IF
```

and about ten times faster than RECOL!

As usual, the DIY project includes full source code in 68000 assembly language, plus a simple hex loader to generate the code for those who do not wish to use an assembler. The first part of the source code appears in **Listing One.** Next month I shall conclude the listing and explain exactly how it works.

Listing Two loads the machine code from DATA statements, using the customary DIY Toolkit hex loader. If you have typed this in before you can re-use the first part of your old file, substituting the new DATA statements from line 590 onwards.

When you RUN listing two, it checks the DATA and stores code in memory. If all is well it asks you for a file name, and saves the DIY Windows code to your chosen device - for example, you might specify FLP1_WINDOWS_CODE. Later you can re-load the file and link the new commands intoSuperBasic,likethis: B=RESPR (676) LBYTESFLP1_WINDOWS_CODE,BCALL B. You may like to add these commands and the CODE file to your BOOT disk or cartridge, so DIY Windows are loaded whenever you re-boot the machine.

If you do not wish to re-type the listing you may be glad to hear that complete source and binary code for this project is available on disk, with comprehensive documentation and example programs. The DIY Toolkit disk includes heap memory commands from *QL World* October 1989, example programs to save and reload window contents on any device, and a routine that will move any window around the screen, under cursor control.

Ask for DIY Toolkit Volume W, which costs £7 on its own, or £3 if you order other DIY Toolkit volumes at the same time. Write to Richard Alexander at DIY Toolkit, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA, or call (0559) 384574. DIY Toolkit files are also available on microdrive tape; please send one blank cartridge for each volume required.

The DIY Windows package includes seven new SuperBasic commands and functions, including one 'alias' to reduce problems caused by inconsistent English spelling. They all expect a mandatory channel number as first parameter; this should be the number of an open SuperBasic SCR or CON channel.

The keywords report 'channel not open' if the channel is closed or has never been opened, and 'bad parameter' if the parameter is not a display channel. A hash character is not compulsory but should preceed the channel number, for consistency with other QL commands.

The function to save a window in memory is W_STORE. This takes a Basic channel number as parameter and returns the address of the memory where it is stored. An 'out of memory' error is reported if there is insufficient memory to hold the display in the Common Heap.

W_CRUNCH is the equivalent function

```
430         POKE start+byte,16*hb+lb
440         checksum = checksum + 16*hb + lb
450         byte = byte + 1
460     END FOR b
470 END REPeat load_hex_digits
480 READ check
490 IF check <> checksum
500     PRINT"Checksum incorrect. Recheck data.":STOP
520 END IF
530 PRINT"Checksum correct, data entered at: ";start
560 END DEFine HEX_LOAD
570 :
580 REMark Space requirements for the machine code
590 DATA 676
600 :
610 REMark Machine code data
620 DATA "43FA025034780110","4ED26100009E5543"
630 DATA "660001 5C3C31E802","54AE005860 0C6100"
640 DATA "008A534366000148","7C0061000 10C3E00"
650 DATA "74FF70184E414A80","666E22063007217C"
660 DATA "42756661FFFC20EE","00E02D4100E020C5"
670 DATA "30C430C655466B0A","6602528C47FA01A6"
680 DATA "600447FA01E66100","007A38012A01383C"
690 DATA "081FD28169145344","2A0170102205E1A1"
700 DATA "690498402A01E240","66F272063078011A"
710 DATA "4E90226E00585D89","2D4900582385E802"
720 DATA "3384E8007802700 0","4E75347801124E92"
730 DATA "666A3E31E8006B5E","54AE0058CEFC0028"
740 DATA "DEAE0030BEAE0034","6C502E3678006B4A"
750 DATA "20474E7561D45343","660000946 15A47FA"
760 DATA "0 15C45EC00803400","4E93284A51CCFFF4"
770 DATA "70004E7561B45543","66743C31E8026138"
780 DATA "4A466B0C528C6702","7CFF47FA010260D2"
790 DATA "47FA011060CC70F1","600270FA588F4E75"
800 DATA "6188554366483C31","E802610C4A466AD6"
810 DATA "528C47FA00E460AA","45FA008C76FF7009"
820 DATA "4E434A8066D63001","2801284948 41C2C0"
830 DATA "4A466702E2895C81","5C812A0148443004"
840 DATA "E248534048445344","4E7570FC4E7570F1"
850 DATA "4E7549EB0010BBCC","66F4518D6100FF2C"
860 DATA "264D508D34780118","4E9266E42E31E800"
870 DATA "6BDC0807000066D6","24473C2A000A6198"
880 DATA "20475888BA9866C2","B85866BE54885546"
890 DATA "6B0C6602528C47FA","0072600 0FF2647FA"
900 DATA "00746000FF1E7028","B0906700FFA23028"
910 DATA "0018E44808800000","22680032D2C03228"
920 DATA "001AC2FC0080D3C1","7206D2680018D268"
930 DATA "001CE4490881 0000","9240484132280 01E"
940 DATA "70004E7510DC528C","51CAFFFA4E751886"
950 DATA "548C51CAFFFA4E75","1914568C51CAFFFA"
960 DATA "4E7518DC51CAFFFC","4E7518D8528C51CA"
970 DATA "FFFA4E7538D851CA","FFFC4E75121C1954"
980 DATA "FFFF18C151CAFFF6","4E7530DC51CAFFFC"
990 DATA "4E750005FF1E0657","5F53484F5700FE76"
1000 DATA "06575F53574F5000","FE6C06575F535741"
1010 DATA "5000FE8207534554","5F524544FEA40953"
1020 DATA "45545F4752454 54E","00000002FD920757"
1030 DATA "5F53544F5245FD74","08575F4352554E43"
1040 DATA "48000000","*",64709
```

to save a window in compressed form. The compression scheme is simple but effective, halving the memory requirement. It takes advantage of the way that QL screen memory is organised. The Mode4 screen occupies 32K of memory, using 128 bytes per line of 512 pixels. Alternate bytes are used to hold red and green information for each group of eight pixels. Even bytes record the colour green, and odd bytes the colour red, so POKE 131072,255 plots a green line eight pixels wide at the top left corner of the display, assuming that area was black initially.

If both red and green bits in adjacent bytes are set, the display is white, so POKE 131073,15 turns the four rightmost pixels of the green line to white. 15 is 00001111 in binary notation, allowing one bit per pixel.

W_CRUNCH saves memory by storing only the odd or even bytes from a window. It takes two parameters and returns an address, like W_STORE. The first parameter is the channel number, followed by the code of the colour to be stored: as usual, use 2 for Red or 4 for Green. Thus X=W_CRUNCH(#1,4) sets X to the address where the green bytes from window #1 are stored.

Most QL windows contain text in two colours, such as black on green, or white on red, and this compression scheme works well in such cases. You only need to store the bytes that differ between ink and paper - green, to suit these examples - and set the appropriate background before you restore the window.

Greater compression may be possible by recording the 'run length' of sequences of particular colours, so blank lines can be encoded as the colour, and the number of pixels of that colour. Such an approach gives good results if the window contains areas of solid colour, but may be counter-productive if the image is complicated or uses stippled colours.

The amount of shrinkage is hard to predict when sequence compression is used. In the worst cases the squashed image may need MORE memory than the un-compressed original, because of the extra bytes recording the length of short runs! Sequence compression could be added to DIY Windows, but it would slow things down; in general W_CRUNCH is preferable because it gives fast, predictable savings.

Stored information can be re-displayed at high speed with W_SHOW #1,X. This works with four-colour images recorded by W_STORE, or two-colour information generated by W_CRUNCH. The memory at X remains allocated, so you can W_SHOW the window again once it has been stored.

When you have several active windows you end up using quite a lot of memory to store their contents, so it is important to release that memory when it is no longer needed. DIY Windows makes this easy, especially if you have Toolkit 2 on your system.

Each area of memory used by W_STORE and W_CRUNCH is linked into Toolkit 2's list of temporary allocations, normally used by the ALCHP function. Such allocations are automatically released when you type LOAD, LRUN or NEW to load a new program. The linkage means that these commands also de-allocate memory used by DIY Windows.

You can also release the memory with the CLCHP command, or use RECHP to de-allocate a specific area. The first four bytes of the area are used to extend the ALCHP list, so the command to release window details at X is RECHP X+4. Toolkit 2 reports a 'bad parameter' error if you supply an inappropriate address.

All is not lost if you do not own Toolkit 2, as window storage space can also be released with DEALLOCATE from Turbo Toolkit, or DIY Toolkit's own DISCARD command. These Toolkits do not automatically release memory when a new program is loaded. They expect you to specify the same address used with W_SHOW, such as DISCARD X or DEALLOCATE X.

DISCARD checks that the parameter is valid, while DEALLOCATE just passes it on to Qdos and hopes for the best! The heap may be corrupted, causing the machine to crash, if you get it wrong.

The memory allocated by W_CRUNCH and W_STORE is owned by the task that calls the function. This means that it is automatically released when a compiled SuperBasic task terminates. You can ästill use RECHP, DEALLOCATE or DISCARD to free space as you go along.

Each window memory allocation has a 12-byte header, followed by the information from the display memory. The first long word is the link used by Toolkit 2, followed by the total length of the area,

including the header. Next come two words: the height of the stored window in pixels, and a flag to signal W_CRUNCH compression, set to zero for W_STORE, and two or four otherwise.

When a crunched image is restored with W_SHOW, only the odd or even bytes in the window are affected. Unless you are wearing a pair of 3D glasses, you should clear the other bytes with CLS or one of the other commands in DIY Windows, or you may end up with a disconcerting mixture of two displays, one red, the other green!

W_SHOW reports 'bad parameter' if the supplied address is odd or negative, and 'out of range' if the stored details will not fit the window. You can store an image and redisplay it in another window, or move the original window to a new location before calling W_SHOW, but the size must be unchanged.

All these routines adjust the window coordinates to fit a word boundary in video memory. This means that they store pixels from a column number evenly divisible by eight, to the next similar boundary at the right hand side of the window. Overlapping windows work best if they are a multiple of eight pixels wide and offset from the left margin by a similar multiple: 0, 8, 16 etc.

This also ensures top speed for text output in the window, particularly if CSIZE 1,0 is selected. The default SCR and CON windows are appropriately placed, as are the (F2) TV windows and the (F1) Monitor mode ones, if the border is ignored. You should take account of any border when calculating the leftmost column; for instance, BORDER 2,2 adds four pixels of red border at the left and right edge of the window.

The other DIY Windows commands allow manipulation of the red and green bytes in display memory. W_SWAP exchanges the red and green bytes in a window, so white text on black is unchanged, but red ink becomes green, and vice versa. There are two valid spellings of the word SWAP, so the command W_SWOP is also accepted and has the same effect. Only one parameter is needed - the channel number.

SET_RED takes two parameters: a channel number, followed by a number indicating the operation to be performed on the red bytes in display memory. SET_RED #0,0 clears all the red bytes, turning a red image on white paper into black on green. SET_RED #0,1 has the opposite effect, setting all the red bytes.

The third possibility copies from the green bytes to the red bytes alongside. SET_RED #0,-1 converts green on black text into white on black. Every green pixel has the corresponding red bit set, changing its colour from green to white.The same three options are available with the SET_GREEN command, but this time it is the green bits that are affected. SET_GREEN #1,0 clears the green bits, SET_GREEN #1,1 sets them, and SET_GREEN #1,-1 copies from red to green.

The commands can process windows generated by other tasks. The easy way to do this is to set up a new window with the same dimensions, from Basic, and use that channel number to save or re-colour the window when it is on the screen. Alternatively you could extract the channel ID from the Qdos table, as demonstrated in DIY Toolkit Volume Q, and tell Basic about it by updating the Channel Table with BPOKE_L, from DIY Toolkit Volume B.

The August DIY Toolkit will conclude listing one and discuss possible modifications to the code to make best use of Minerva, Mode8 and available memory. The article will also explain a simple way to add icon menus for QL tasks, extending DIY Windows in the direction of a full 'WIMP' system.

SUBSCRIBE TO SINCLAIR QL WORLD!

SUBSCRIPTION FORM

Rates (12 months)

U.K.	**£23.40**
Europe:	**£32.90**
Rest of World	**£40.90**

Select back issues available on request. Please call for details.

Cheques should be made payable to Arcwind Ltd.
Please debit my credit card/Access/Visa

Card No. _____

Expiry Date _____

Send the completed form to:

Arcwind Ltd
The Blue Barn,
Tew Lane
Wootton
Woodstock
Oxon OX7 1HA
Tel: 0993 881181
Fax: 0993 811481

Please send me a 12 Month subscription to Sinclair QL World

Name _____

Address _____

_____

Telephone No. _____

Signed _____

Date _____

SOFTWARE FILE

PUDGE

Program:	Pudge
Memory:	128K
Publisher:	CGH Services, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA
Price:	£12.50 (disk or mdv)

Rich Mellor gets a Naggon in this alien-zapping game.

This is a brand new arcade-puzzle game written by Damon Chaplin, following his earlier successes, *Speedfreaks* and *Assault & Battery* (also now published by CGH Services – see *QL World* December 1990). QL owners will be pleased to learn that this game (like the current versions of the other two) is not copy protected, and works 100% on a Minerva rom.

The program is supplied with a short manual which explains a little of the background of the game and how to play it. Apparently the original idea for this game was based upon a similar public domain game for the Atari ST, although I get the feeling that the scenario was very much Damon's own work.

Cartoon

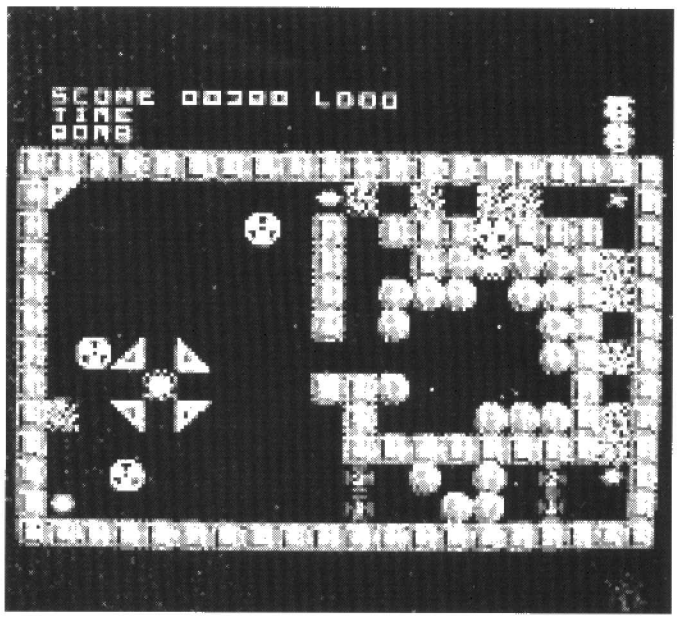
The idea behind the game would at first seem simple, you control a little cartoon-like character named Pudge, who is living on a planet which is doomed to destruction due to nuclear dumping. Pudge is an environmentalist who seems to have become trapped on this planet (the environmental issues on Earth must have seemed too easy for him), and wishes to save it. However, many things stand in his way: radioactive earth, nuclear canisters, bombs and Naggons (the race of beings responsible for the dumping). It is Pudge's job to clear away the radioactive material and defeat the Naggons, thus ridding the planet of its terrible fate (at least until a sequel is devised).

Once loaded, you are presented with the first of 32 screens, each of which is made up of various colourful symbols representing the different nasties that Pudge has to clear away, as well as various other items to help Pudge in his task. You merely have to guide Pudge around the screen by using the cursor keys (or a joystick) in order to achieve your various tasks. Collecting the radioactive earth seems easy – you merely have to move Pudge over this – but then nothing is ever that simple as you have to work quickly to avoid radiation poisoning. Luckily there are various anti-radiation pills scattered around the screens that you can collect which will give you more time to spend achieving your tasks.

Before you can proceed to the next level, you have to destroy the radioactive canisters which litter the planet – by pushing them into one of the 'chompers', which you will find on the screen. This is in fact fairly easy, since any number can be moved at a time, by merely placing Pudge behind the object and then moving Pudge in the required direction with the object in front of him. Triangular parts of the walls allow you to turn the direction in which these canisters are being moved quite easily. The canisters should not be destroyed too early on all of the screens however, since they have another use which I shall come to later.

Your time is however limited by the bombs which appear on most the screens. To destroy one of these bombs, you must surround them by four boulders (forming the shape of a cross). Boulders are quite heavy and Pudge can only move one at a time, so it is important to plan your moves carefully; it is all too easy to trap Pudge by moving one boulder behind another and making it difficult for him to push it. It is here that the first puzzle element comes into play, and although it can take some time to plan your strategy for moving a set of boulders, even young players should be able to come up with some sort of a solution early on in the game (luckily the game can be paused by pressing F2 at any time to allow you to sit and have a long think – unfortunately the manual did not mention this).

On later levels, you will come across the Naggons who move around the screen in search of you – touching them means instant death for Pudge, and so it becomes a major priority to destroy these. Naggons can be destroyed in a similar way to the bombs, ie by surrounding them with four radioactive canisters. Unfortunately the main problem here is not how to get the canisters to the Naggons, but rather how to set up a trap before finally placing the final canister. Again, quite a lot of thought will be needed here, especially on one of the screens where there are three Naggons to destroy and only four canisters (luckily there is no bomb to take care of on this screen).



Overall the game is extremely addictive, but beware: on some levels there is only one way in which to complete the screen within the allowed time. One early mistake may prevent you from completing the given screen, as you are blocked in. But to help players out a little, each screen has a distinct password, displayed at the top of the screen, and typing this in on the title screen allows you to start the game from that screen – so you need only practice those screens that you are having difficulty with.

Diagonal

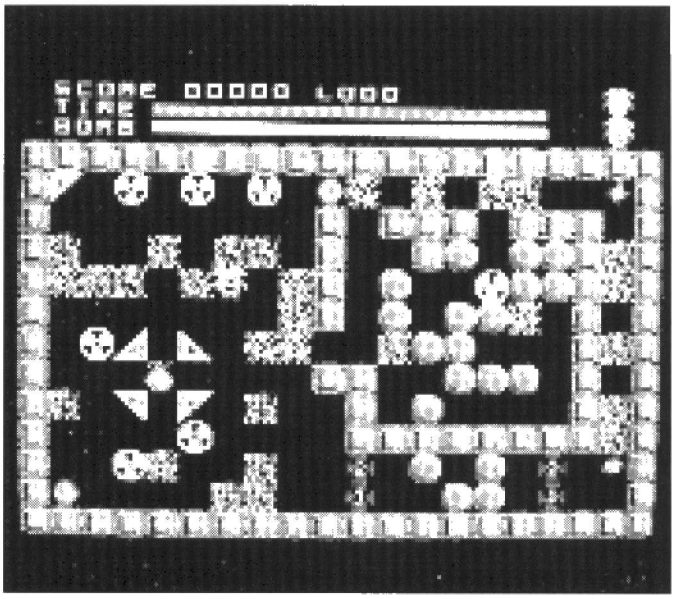
Those of you with full sized keyboards will be pleased to learn that the game does not entail any need for diagonal movement. If you are like me, you will have soon found out that a full sized keyboard tends to make it difficult to play games where you need to press two keys at a time. (When you are holding down two keys and release one, the QL does not tend to recognise that any key is being pressed, still – I wonder if this is just because of the Jurgen Falkenberg interface, or if there is a fix?)

In all there are 32 original screens to complete, although I have only managed to reach level 12, despite many many hours of play. The level of difficulty is just right to keep you coming back to the game, and for those of you who are clever and lucky enough to complete all 32 screens, a screen design program is provided to enable you to create your own screens. This appears to be well written but does not check to see if a screen can actually be completed, and so only those players who have managed to complete all of the supplied levels should really try this one.

Slow down

About the only problem of which I am aware is that the game runs too quickly on Miracle's *Gold Card*, although I understand that there is now an extra command on the Gold Card which allows you to slow down the speed of keyboard access for games such as this.

This game is extremely well written and is possibly the first environmentally friendly computer game for the QL and as such deserves success (although I wonder just how close to the Earth's future this could be?). The screens are all well designed and wonderfully colourful and should attract even the youngest games players to the QL. What is more, this program is useful for developing strategic planning and should suit all types of QL users, from those who would rather have a game of chess to those who find great pleasure out of zapping aliens.



CARE ELECTRONICS

QPAC II by Tony Tebby

MAKES YOUR QL TRULY MULTI - TASKING

QPAC II Main menu windows adjust automatically to size. Files, directory, view, print, delete, backup, jobs, pick, Rjob, sort, channels, things exec, wake, buttons, Hotkey, Hotjobs. Fully multitasking, allows many programs to run at once. Requires min of 256k expanded memory ..... SPECIAL OFFER ONLY £39.48b

Upgrade QRAM to QPAC II return master .. .. @29.61b

PLEASE SEND SAE FOR A COPY OF THE QPAC II REVIEW

QPOWER REGULATOR

Switch mode power supply, QL runs cool, stops lock ups (due to voltage drop). Helps filter noise out. Simple to fit (no soldering) £25.85c

QL SUPERTOOL KIT II by Tony Tebby

Over 118 Commands: Full Screen Editor, Key Define Print Using, Last Line Recall, Altkey, Job Control, File Handling, Default Directories, Extended Network.
16k Eprom Cartridge Version ..... @£23.50d
Configurable Version of Microdrive ..... @£23.50d

ZITASOFT SOFTWARE by Steve Jones

LOCKSMITHE copies M/DRIVE - M/DRIVE ..... @£14.10c
4MATTER + LOCKSMITHE copies M/DRIVE - DISC ..... @£23.50c
SIDEWINDER - High resolution printer driver prints full screens or parts of screens from postage stamp size to large banners. Prints sideways, invert, scale, mirror, text insertion ..... @£20.21c
SIDEWINDER PLUS - (for expanded QL's) includes all the features of above, PLUS multiple label printing, desktop publishing files and printer drivers for The LC10 JX80 24 pin and colour printers (Please state 3.5" disc or M/Drive) ..... @£23.97c
Upgrade to Sidewinder Plus ..... @£8.46c

SOFTWARE 87 (State MDV or Disc)

TEXT87 Plus 4 ..... @ £79.00b
FOUNTED 89 ..... @ £15.00
FOUNTED 88 ..... @ £25.00c
248B PRINTER DRIVER ..... @ £15.00c
Upgrade to Text 87 Plus 4 from V3.00. Return master £39.00b
Upgrade to Text 87 Plus 4 from all other versions. Return Master £49.00b
Send SAE for further info on above.

MIRACLE SYSTEMS PRODUCTS

0K Disc Interface (inc Tool Kit II) ..... @ £99.64b
QL Centronics Printer interface ..... @ £29.61d

QL HARDWARE

QLKeyboard Membrane ..... @ £11.75d
Care Eprom Cartridges each ..... @ £6.11c
ULA CHip ZX8301 ..... @ £15.98c

READY MADE LEADS

RGB QL TO PHONO £6.11c
RGB 8-6 PIN £7.52c
RGB 8-7 PIN HITACHI £7.52c
RGB 8-7 PIN FERGUSON £7.52c
RGB 8 PIN TO SCART £11.75c
RGB 8PIN TO CM8833 MK II £11.75c
6WAY PCC TO 25WAY D £9.87c

TONY TEBBY SOFTWARE (QJUMP)

QTYP Type/Spell Checker ..... @ £30.55d
QPAC 1-Desk top accessories, calendar, alarm, calculator, typewriter, digital clock sysmon ..... @ £22.56d

CARE ELECTRONICS
DEPT QL, 15 HOLLAND GDNS
GARSTON, WATFORD,
HERTS, WD2 6JN.
Tel: 0923-894064
Fax: 0923-672102

Please add carriage
a=£6.11 b=£3.76 c=no charge d=£2.35

HOW TO ORDER

By Post. Enclose your Cheque/PO made payable to CARE Electronics or use ACCESS/VISA. Allow 7 days for delivery

QUANTA

Independent QL/Thor Users Group

Worldwide Membership is by subscription only, and offers the following benefits:
Monthly Newsletter – up to 40 pages
Massive Software Library – mostly FREE!
Free Helpline and Workshops
Regional Sub-Groups. One near you?
Advice on Software & Hardware problems
Discounts from most major suppliers
Subscription just £14 for UK members
Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

* Now in our NINTH successful year *

Further details from the Membership Secretary

Bill Newell
213 Manor Road
Benfleet
Essex. SS7 4JD
Tel (0268) 754 407

SOFTWARE FILE

QUICKDISK + DISKTOOL

INFORMATION
Program: QuickDisk+DiskTool
Price: £19.95 plus 5% to Europe, 10% to rest of world.
Supplier: Digital Precision Ltd., 222 The Avenue, Chingford, London E4 9SE.
Tel. 081 527 5493

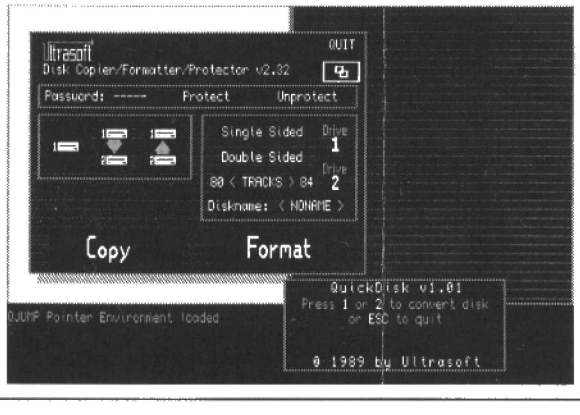
Bryan Davies tries two updated disk-handling facilities which started life with Ultrasoft.

These two utilities are part of a set which originated from Martin Berndt of UltraSoft, in Germany, and is now part of the Digital Precision collection. DP have gone through these programs, making changes where thought necessary, and re-writing the instructions. Both programs work under the QJump *Pointer Environment*, which is supplied on the disk. Users who already have QPac installed should not need to load the PE files, although they may find that the ones supplied are later versions than their own.

The programs are supplied only on disk. A 4-page instruction leaflet is supplied; this is adequate to describe the program functions, but there is little advice on using the Pointer Environment, on the basis that it 'requires little documentation', which is questionable.

Load check

As is usual with DP products, copy-protection is not employed and a working copy can be made by using the COPY or WCOPY command. Loading is via a boot file, and a check is made to see whether or not the PE has already been loaded, to avoid trying to load it twice. Both programs are automatically EXECed from the boot, but the latter can be altered so that one or both are not EXECed straight away. They can be started by *TaskMaster*, *QRam*, *QPac*, etc. DiskTool can be defined as a Hotkey, if you use Hotkey II, or it can be put onto a 16 KB eprom if you have (access to) an eprom blower (two files are supplied for these purposes). The standard QJump CONFIG routine is supplied, for configuring the programs. You may be able to incorporate the programs into an existing, non-PE boot, but some experimentation is likely to be required, as there might be compatibility problems.



Don't touch

When the instructions get down to describing what the programs do, they hit on what may be a sore point with users (such as myself) who have one-third-height 3.5 in Mitsubishi disk drives. QuickDisk 'speed-optimises disk organisation' and, to do that efficiently, it should be configured for the specific drive controller chip installed in your disk interface. The chances are that most users do not have this information to hand, and you are advised to take the cover off the interface and check to see what is written on the chips. What really ought to be said also is: don't touch the chips. Literally. The drive controller chips can be damaged simply by being touched with the small amount of static electricty passed by one finger, even if you can't feel it.

What you are looking for is the designation WD1770, or WD1772, on one of the larger chips. Early interfaces of most makes will usually have the WD1770. The WD1772 has been fitted to Trump Cards, and that is the best guess if you don't feel inclined to risk taking the cover off your TC. The WD1772 is nominally 'better'; unfortunately, it does a poor job of handling the aforementioned Mitsubishi drives. For this reason, some Trump Cards will have been retrofitted with the WD1770. (If you have a Trump Card and these drives, and the latter sound as though they are being shaken to bits, you may need a 1770!) The default setting of QuickDisk is for the 1772; fortunately, it should work (albeit not at peak speed) even if set wrong. It seemed to work satisfactorily with either setting on the review system. The warning is given to back up important files before proceeding with use of QuickDisk, just in case something goes wrong. Later in the instructions, it is pointed out that recovery of files on a damaged disk may be more difficult (with Media Manager SE, for instance) once the disk has been treated by QuickDisk.

Quick files

The action of QuickDisk is to 'convert' formatted disks in such a way that files copied onto them are accessible more quickly.

When the boot is run, DiskCopy and QuickDisk come up in their own windows, the latter having the cursor. Be careful with the ESC key as it has the usual, final effect on programs running under the PE - they get killed. Use CTRL-C to switch to other programs if you want the two utilities to stay alive. QuickDisk offers the simple choice of pressing 1, 2 or ESC. The 1 or 2 key initiates conversion of the disk in flp1_ or flp2_, the process taking a few seconds. The disk should be both formatted and empty before being treated. A DIR subsequently will show no change. If the FORMAT command is used on the disk, the latter will revert to its normal, unconverted state.

That's all there is to QuickDisk, so the question is 'what good has it done me?' The first thing to be assured of is that it has done no harm;

normal operations, such as WCOPY, involving converted disks, are carried out the same as normal. There is no evident difference. Using WCOPY to copy a mixed bag of 52 files (occupying 717 KB when copied) from win1_ to flp1_, until the 'drive full' message appeared on the screen, took 4 minutes 58 seconds with a normal, unconverted disk. A DIR (three-piece) of flp2_ then took 10 seconds. A WCOPY of all files from flp2_ to ram1_ took 5 minutes 58 seconds ('drive full'). Deleting all files on flp2_ with WDEL took 33 seconds. The same operations after the disk had been treated by QuickDisk took 3 minutes 10 seconds, 5 seconds, 5 minutes 18 seconds and 25 seconds. So, housekeeping operations are performed appreciably faster. As a check of typical word-processing operations, a 312 KB text file was loaded into Perfection; it took 2 minutes 35 seconds from a normal disk, 2 minutes 23 seconds from a converted disk. Saving the same file took 57 and 40 seconds, respectively. Likewise, appreciably faster.

You couldn't wish for anything much simpler than QuickDisk, and DiskTool is not complicated either. The program window offers its services as 'copier/formatter/protector', and a sub-window displays icons for an flp1 to flp1 copy, flp1 to flp2, and flp2 to flp1. A further sub-window displays the options single- or double-sided, 80- or 84-track, disk name, drive 1 or 2. Copy and Format each have their own sub-window, and Password and its three options, (sic; see lower down), Protect and Unprotect, are together in another.

Cursor

As usual with the Pointer Environment, the cursor changes appearance as it is moved around the screen, starting off as a floppy disk symbol with QL written on it. Cursor navigation by mouse (Qimi-type) is much faster than by cursor keys, but this program doesn't place great demands on the navigator and is not one where a mouse is virtually essential. Movement of the cursor into the area of an option causes a highlight box to appear around the option name. A list of keys is given for direct activation of options; several of them are mnemonic and easy to memorise (eg N for Disk Name, 1 for drive 1). Standard for PE operations, but likely to be confusing for non-devotees, is that options are generally selected by pressing Space rather than Enter, but some options - such as Copy and Format - require Enter rather than Space.

Having pressed Space on all the appropriate options to do an 80-track, double-sided format, it was a bit disconcerting to find 84-track highlighted during the operation, and then have a box pop up after some seconds with a couple of exclamation icons, and the messages 'error' and 'format failed'. The change of tracks actually occurred because of my hitting Space when 80-track was already selected; this automatically changed the setting to 84-track - something doing my thinking for me and coming up with the wrong answer. The instructions tell you to ignore the error messages! Returning to SuperBasic and doing a DIR confirmed that the disk had been formatted, and to 1512 sectors, as stated. That is, 72 sectors (36 KB) more than standard. Usable sectors are 1506, against the standard 1434. The instructions don't explain why the program bothers to display the misleading messages.

Difficult

As with QuickDisk, you are warned that the sorting-out of a corrupted 84-track disk may prove more difficult than for an 80-track one.

Having gained the extra space, it makes sense to go one step further and apply the QuickDisk conversion to 84-track disks. It is stated that there is no penalty from either operation, apart from that mentioned about recovery from disk problems. The 'treated' disks should be read from, and written to, by SB and programs, without difficulty, and in other drives too; there is no need for the PE or either of these programs to be present subsequently.

One operation produced undesirable results - formatting a disk 84-track, single-sided. Although the format seemed to be performed alright, a subsequent WCOPY to the disk resulted in some pained noises from the drive and a string of 'bad or changed medium' messages which could be stopped only by resetting. Disk and drive seemed in order afterwards, though. Bearing in mind the points made in the instructions about disk controller chips, the standard QJump CONFIG routine (supplied on the disk) was used to change the default chip setting for QuickDisk from WD1772 to WD1770, as the latter type is installed in my Trump Card (because of the aforementioned problem). This change doesn't apply to DiskTool. The WCOPY operation still gave the same result, and QuickDisk hadn't been used this time, so it appears best - on my system at least - to stay away from the single-sided format. There was no problem with a WCOPY after a double-sided 84-track format, and conversion.

Disk name

The Copy icons are simply a convenience - there is nothing special about the copying process. An advance on SuperBasic is provided by the Disk Name option, which can be used at any time, not just when a format is done as is the case with SB. A small feature, but one which many users must have wished for many times.

The Password function could be useful if you use your QL in an office environment where there are confidential files which some people should not have access to. The protection works on both program and data files; it is necessary to have both a copy of DiskTool and the password to be able to access protected files. Attempts at access without these things result in 'bad or changed medium' messages, or '0/0 sectors' when DIR or STAT commands are used. Copying, loading or viewing operations just fail completely.

The use of a password is not without pitfalls for the unwary and you are warned that attempting to unprotect a protected disk and giving the wrong password in the process will make everything on the disk permanently inaccessible, after only one attempt. A rather drastic action, since it is quite possible an 'authorised' user would make a slight slip when entering the password. Backups are again the order of the day.

Password

The instructions could have been a little clearer for such a sensitive operation; the box shown with —— in it in the illustration is for entering the password, and you have to place the cursor on it first, then press Space. This changes the box contents to ????, if no pass word has been entered. Surprisingly,, this action displays the password if you have entered one previously during the current session, so don't leave the DiskTool window on the screen for anyone else to play with. It might have been more self-explanatory to have this box named Password. The password (up to four characters, case-dependent) is typed in and Enter pressed. You then select the Protect box and press Space.

As claimed, the contents of a Protected disk became inaccessible to all normal operations. When the DiskTool window was again called up to Unprotect the disk, the password was still displayed in it, and it appears that anyone capable of using CTRL-C could discover the last password used at any time until either DiskTool has been killed or the QL reset. Both the Protect and Unprotect operations are accompanied by a message box saying 'disk change', even though there has been no change of disk; you have to press Space with the cursor on an OK box to complete the operation.

The two programs are easy to use and provide operational bonuses which are well worth having. Use of them should be accompanied by extra caution in the matter of making backups. The password function also requires care in use. The price is reasonable and both QuickDisk and DiskTool should make useful additions to a utilities collection, especially for Pointer Environment users.

QL INTERNATIONAL

An International QL Meeting was held in Germany in March. Dilwyn Jones reports.

On the 20th March, a minibus full of Quanta members set out for the International QL Meeting in north-west Germany. The venue was a school hall in Münster-Roxel. This was the latest in a series of outings organised by Quanta, the QL user group. The International QL Meeting was organised by Franz Hermann of the Sinclair QL User Club eV of Germany.

Several German traders attended the meeting, including Jochen Merz, Albin Hessler Software and Jurgen Falkenburg Computer Teknik. Jochen Merz demonstrated and sold his well-known products for both the QL and the Atari ST. The ST-QL emulator card is well known and widely used in Germany - it offers a very high degree of compatibility with QL software. As a software supplier, he is widely respected for his range of Pointer Environment software, including several of Tony Tebby's products.

Albin Hessler showed his *Easy-Ptr II* system for Pointer Environment programmers. This simplifies the writing of menus and so on, removing the need for the more complex programming involved when using the Qptr package by itself. Albin also demonstrated a PC-type serial mouse connected to a QL via the QL's SER ports. With the serial mouse driver software, it enabled the mouse to drive the Pointer Environment as though it was a Qimi mouse (the original QJump mouse system for the QL). This will be available in Britain, eventually, through EEC Ltd., I was told.

Jurgen Falkenburg produces a range of hardware add-ons for the QL, including a PC keyboard interface, bus expander, EAN barcode reader, QL picture scanner and an A/D converter. His products are not that well known outside of Germany, which is a pity since he has obviously put a lot of work into them. Traders and user groups from other countries were present too, including Progs of Belgium, Ergon Development from Italy and Cowo Electronic from Switzerland. I met QL users from places as far apart as France, the Netherlands, and Scandinavia.

While exploring the various stands, I came across two very interesting pieces of software. We all know that Sinclair Research produced other computers before the QL, but did you know that the QL can run ZX81 and Spectrum software? I saw the ZX81 emulator

Photos by Bill Newell

DILWYN JONES HARD AT WORK.

BILL RICHARDSON OF EEC IN FULL FLIGHT.

while at the show. It is a software package written by Carlo Delhez, a QL user in The Netherlands (a demonstration version is available from Qubbesoft in Britain). He has also written a Spectrum emulator and I was told that there are at least two other such emulators being developed.

Language proved to be no barrier at the meeting - people spoke whichever language they could best communicate in! The QL proved itself to be an international medium of communication. One amusing experience occurred when two users from different countries tried to speak English to each other, but had to grab the nearest 'Englishman' to translate one English accent into another, except that, by choosing me, they got one with a Welsh accent to confuse them even more!

QUANTA - WHAT? ONLY £8 FOR JAN JONES? DONE!

Most of the British traders who advertise in QL World were present (notable exceptions being Digital Precision and CGH Services). EEC, Quanta, TF Services, Miracle Systems and Qubbesoft were among those present. Miracle Systems apparently did a roaring trade in Gold Cards, which are now much reduced in price at £225, compared with nearly £400 when first launched.

Several talks, lectures and demonstrations took place. Subjects included Minerva, bulletin boards, *Fleet Tactical Command* and the future of Qdos and the QL. Sadly, I missed most of these because I did not know what was on at which time, a point to improve on for next time, maybe. The whole show was conducted in a friendly, informal manner which brought QL users from several countries together for an enjoyable day out in a sea of Sinclair QLs and ST-QLs. Some no doubt departed poorer, having spent their money on bits and pieces for the QL, while others shared knowledge and made new friends.

After a very enjoyable day at the International QL Meeting, we retired to a local hotel for the night, before departing for a very wet journey back to Britain the following morning. I, for one, am certainly looking forward to the next such outing.

a·C·L·U·B·s

FRANCE

QL Contact France. Contact: Jean-Louis Dianoux, 38-40 Rue Stephenson, 75018 Paris, France.

BELGIUM

Club Sinclair BruQsL (Belgium) Contact: Jaques Tasset, Aarlenstraat 104, 1040 Brussels, Belgium

SWEDEN

International QL Conference bulletin board system (Swedish and English). Contact: Michael Cronsten, System Operator, Jamten-TCL, S Soere 1073, 83030 Lit, Sweden.

USA

New England Sinclair QL User Group (USA) Membership Secretary: Sherm Waterman, PO Box 8763, Boston, MA 02114 8763, USA. Magazine: *NESQLUG News*. Editor: Peter Hale, 195 Central Ave., Chelsea, MA 02150, USA.

NORWAY

Norwegian All Sinclair Association (NASA) Contact: P Monstad, NASA, N-5580 Oelen, Norway. Magazine: *Sinclair Magazine*.

ITALY

Qitaly Club Chairman: Roberto Orlandi, Via Brescia 26, 25039 Travagliato (BS), Italy. Tel. (local) +39 30 6863311. Magazine: *Qitaly Magazine*. Editor: Dr Eros Forenzi, Via Valeriana 44, 23010 Berbenno (SO), Italy. Tel. (local) +39 342 492323.

TURKEY

QL Qlub (Turkey). Contact: Bulent Artuz, Prof. Sitesi B/1 D/5, Etiler 80600, Istanbul, Turkey.

AUSTRIA

DER Computer Club. Contact: Peter Postl, Stiegerg 5, 1150 Wien, Austria

ENGLAND

Quanta (UK) Membership Secretary, Bill Newell, 213 Manor Road, Benfleet, Essex SS7 4JD. Magazine: *Quanta*. Editor: Bill Fuggle, 20 Widnes Avenue, Selly Oak, Birmingham B29 6QE.

Bristol sub-group: Roy Brereton, 94 Teignmouth Rd, Clevedon, Avon BS21 6DR.

Essex sub-group: Dave Walker, 22 Kempton's Mead, Potters Bar, Herts EM6 3HZ.

London sub-group: Jeremy Davis, 6 Elmcroft Crescent, Harrow, Middlesex HA2 6HN.

Northern Ireland: Billy Turkington, Fairyhill, Rostrevor, Newry, Co., Down BT34 3BB.

Mid-Southern sub-group: Geoff Fish, 44 Billing Avenue, Wokingham, Berks RG11 4JE.

QL User Group (West Midlands): Mike Bedford-White, 16 Westfield Road, Acocks Green, Birmingham B27 7TL.

QL MUG: 3 Barnard Road, Birkenhead, L43 1TT.

SCOTLAND

Scottish QL Users Group Contact: Alan Pemberton, 65 Lingerwood Rd., Newtongrange, Midlothian EH22 4QQ. Newsletter.

HOLLAND

Sin_QL_Air (Netherlands) Membership Secretary: Bob Visser, Snelrewaard 6, 2904 SN Capelle, a/d Ijssel, Netherlands. Magazine: *Quasar*. Editor: C H M Biemans, Elzenstraat 5, 5461 CL Veghel, Netherlands.

GERMANY

Sinclair QL User Club eV (Germany) Foreign Contact: Franz Herrmann, Talstrasse 21, d-W5460 Ochenfels, West Germany. Magazine: *Quasar*.

SPAIN

Qliper Editor: Marcos Cruz, Acacias 44, (Monteclaro), E-28023 Pozuelo de Alarcon, Spain. Magazine: *Qliper*.

Systematic Machine Code Programming

In Part 8, Alan Bridewell applies number inputting to the control of screen drawing parameters.

This series has been building up a library of small chunks of Assembler Language code. Each chunk will do a recognisable job within a program, and be fully annotated, so that it will be clear exactly how to join it to other chunks of code to make a program, and what changes may be necessary to make the chunks fit together.

In the last article, we looked at inputting numbers into a program in such a way that it avoided inputting them directly from the keyboard. This would then overcome the problems of trying to get error-free numeric input. This month we shall look at how to apply these ideas in real programming situations.

The classic situation is of drawing designs on the screen, the way CAD programs do. The example we shall take is of drawing ellipses on the screen. However, if you follow through the program, and understand its principles, you should be able to see how to apply the same basic ideas to other drawing problems, for instance the "rubber-band" method of drawing lines. The point is this. The program user does not want to have to input parameters directly from the keyboard to draw the required shape. For a start, the user may not have the remotest idea what parameters are required, never mind about the actual values. Also, the user may only have a rough idea of what actual shape is wanted, until the shape can be seen on the screen. What the user needs is a routine which will, first of all, draw an ellipse somewhere on the screen. Then, by the use of the cursor keys, the ellipse can be moved around the screen, stretched, shrunk, and rotated, until eventually it is in the desired position, size, shape and orientation. It can then be left there ready for the next bit of the design to be added.

Clearly, what our program must be able to do is to use the cursor keys to alter the floating point parameters for the ellipse, rub out the old ellipse, and then redraw the ellipse according to the new parameters. However this does introduce a few complications. Rubbing out old ellipses is simply achieved by putting the window in XOR drawing mode. This means that the drawn colour is XORed with the background. If the background consists of the drawn ellipse, it will simply be rubbed out. A further problem is that each time the old ellipse is to be kept, and a new ellipse drawn, the ellipse parameters need to be restored to their default values. This means keeping a second set of default parameters to copy into the ellipse parameter table each time we wish to start drawing another ellipse.

Listing One is a short 'COPY' routine, which will copy any block of long words from one address to another. SOURCE and DEST are the start addresses of the source block and the destination block. The loop counter must be set to one less than the number of long words to be copied. Blocks as large as 64K can be copied this way, if the MOVEQ is changed to a MOVE.W command, but it is a bit slow for large blocks. However it's ideal for parameter tables.

The program, then, must do the following.

1. First, start the job, open the window, enable the cursor, and put in the XOR mode.
2. Copy the default parameter table to the ellipse parameter table and draw the default ellipse.
3. The program must then wait for a key press.
4. It must next analyse the key press to find out if it is to leave the routine with the current ellipse left in place, or draw another ellipse.

Listing 1

```
; ************************************************************************
;                              'COPY'
; ************************************************************************
; THIS WILL COPY A BLOCK OF LONG WORDS FROM ONE ADDRESS TO ANOTHER
; THE LOOP COUNTER SHOULD BE ONE LESS THAN THE NUMBER OF LONG WORDS.
;
.COPY               LEA.L     SOURCE,A1 ; SOURCE ADDRESS IN A1
                    LEA.L     DEST,A2   ; DESTINATION ADDRESS IN A2
                    MOVEQ     #4,D1     ; LOOP COUNTER
.LOOP               MOVE.L    (A1)+,(A2)+ ; MOVE LONG WORDS
                    DBRA      D1,LOOP   ; DECREMENT AND LOOP
                    BRA       NEXTBIT   ; BRANCH TO NEXT BIT
;
; ************************************************************************
```

Listing 2

```
; ************************************************************************
;                              'MAIN'
;
.MAIN               JSR       FBYTE     ; GET KEY PRESS
                    LEA.L     BYTE,A1   ; BYTE ADDRESS IN A1
                    CMPI.B    #10,(A1)  ; IS BYTE = 10 <ENTER>?
                    BEQ       CLOSE     ; IF SO, CLOSE
                    LEA.L     BYTE,A1   ; BYTE ADDRESS IN A1
                    CMPI.B    #32,(A1)  ; IS BYTE = 32 <SPACE>
                    BEQ.S     COPY      ; IF SO, NEW ANOTHER ELLIPSE
                    JSR       ELLIPSE   ; TO REMOVE OLD ELLIPSE
                    JSR       TEST      ; TEST FOR KEY PRESS
.FIRST              JSR       ELLIPSE   ; TO DRAW NEW ELLIPSE
                    BRA       MAIN      ; AND START AGAIN
;
; ************************************************************************
```

5. If the key press indicates more adjustments to the current ellipse, it must rub out the old ellipse, ready to draw the new one. This involves drawing the ellipse in XOR mode over the old ellipse.

6. It must then analyse the key press to determine what parameter change is required, and make the required change.

7. Next, it must draw the new ellipse.

8. Finally, it must loop back to wait for another key press.

9. If, at any stage, the key press indicates leave the ellipse and start another, the program must loop back to step 2. If we are to leave the routine, the program must close the window, and kill the job.

Clearly, the first and last steps are only needed to make this sample routine into a self contained EXECable program. Normally, it would be a small part of a larger program. In this case, presumably, the program would already be started with the window open, the cursor enabled, and in XOR mode. So when the main program entered this routine, the first step would be to copy the default parameter table and draw the default ellipse. When all the required ellipses were in place, the last step would be to go back to the main program, and would not close the window or kill the job.

Three Stages

The previous program examples in this series have been ones where the steps have followed each other in a simple linear manner. Also, where there have been repetitions of steps, they have been short ones. This program is more complex, and requires a more subtle program technique. For a start, there are three situations involving the 'ELLIPSE' routine to draw an ellipse. There is drawing the initial default ellipse, drawing the background colour ellipse to rub out the old one, and drawing the new ellipse. 'ELLIPSE' is rather a long routine, and to repeat it three times will make the program rather long. Also the three situations do not easily make for a linear approach.

The solution is to put a bit more 'structure' into the program. This is done by making various chunks of code into subroutines, which can be called in a variety of ways from a 'MAIN' routine. Making a chunk of code into a subroutine simply means making it end with an 'RTS' command to return to the calling code, rather than the usual 'BRA NEXTBIT' type of ending. This allows much more flexibility, and will certainly make the program more concise. It does, however, mean that a separate 'MAIN' routine has to be written specifically for this program, and it cannot really be incorporated into the library of routines for general use. The writing of a 'MAIN' routine obviously requires more programming skill than simply putting together chunks of code, which we have been doing so far in this series. Having said that, I must add that writing 'MAIN' routines is not as difficult as writing many of the chunks which make up the library.

Listing 3

```
; ***************************************************************
;                          'JOBSTART'
;
                    BRA.S     START       ; BRANCH TO START OF CODE
                    DC.L      0           ; (THIS IS STANDARD FORMAT FOR
                    DC.W      $4AFB       ; START OF A JOB)
; ---- ALTER CHARACTER COUNT AND JOB NAME ----
                    DC.W      8           ; CHARACTER COUNT OF JOB NAME
                    DC.B      'ELLIPSES'    ; NAME OF JOB
;
; ***************************************************************
;                          'CONSOLE'
;
; ---- ALTER LABEL TO .START ----
.START              LEA.L     PBLOCK,A1           ; PBLOCK ADDRESS IN A1
                    MOVE.W    #C6,A2              ; UT_CON VECTOR IN A2
                    JSR       (A2)                ;
;
; SAVE THE CHANNEL ID WHICH UT_CON ROUTINE LEAVES IN A0.
;
                    MOVE.L    A0,-(A7)            ; SAVE CONSOLE ID ON STACK
                                                  ; * NOTE * THIS CHANGES A7
                                                  ; BY -4, READY FOR NEXT
                                                  ; STACK ENTRY.
;
; CLEAR THE WINDOW
;
                    MOVE.W    #$FFFF,D3           ; INFINITE TIMEOUT
                    MOVEQ     #20,D0              ; #SD_CLEAR IN D0
                    TRAP      #3                  ;
; ---- ALTER BRANCH ADDRESS TO CURSOR ----
                    BRA.S     CURSOR              ; SKIP BLOCK
;
; ---- ALTER BLOCK TO REQUIRED WINDOW ----
.PBLOCK             DC.B      4                   ; GREEN BORDER
                    DC.B      2                   ; 2 PIXELS WIDE
                    DC.B      0                   ; BLACK PAPER/STRIP
                    DC.B      7                   ; WHITE INK
                    DC.W      512                 ; WIDTH
                    DC.W      200                 ; HEIGHT
                    DC.W      0                   ; X POSITION
                    DC.W      0                   ; Y POSITION
;
; ***************************************************************
;                          'CURSOR'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
.CURSOR             MOVEQ     #$E,D0      ; #SD_CURE IN D0 (ENABLE)
                    TRAP      #3          ;
;
; ***************************************************************
;                          'WRITEMODE'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
.WMODE              MOVEQ     #$2C,D0     ; #SETMD IN D0
                    MOVE.W    #$FFFF,D1   ; XOR INK INTO BACKGROUND
                    TRAP      #3          ;
;
; ***************************************************************
;                          'COPY'
;
; ---- ALTER LOAD ADDRESSES TO DEFAULTS AND EXA ----

.COPY               LEA.L     DEFAULTS,A1 ; DEFAULT PARAMS ADDRESS IN A1
                    LEA.L     EXA,A2      ; ACTUAL PARAMS ADDRESS IN A2
                    MOVEQ     #4,D1       ; LOOP COUNTER
.LOOP               MOVE.L    (A1)+,(A2)+ ; MOVE TWO PARAMETERS
                    DBRA      D1,LOOP     ; DECREMENT AND LOOP
; ---- ALTER BRANCH ADDRESS TO FIRST ----
                    BRA.S     FIRST       ; DRAW DEFAULT ELLIPSE
;
; ***************************************************************
;                          'MAIN'
;
.MAIN               JSR       FBYTE       ; GET KEY PRESS
                    LEA.L     BYTE,A1     ; BYTE ADDRESS IN A1
                    CMPI.B    #10,(A1)    ; IS BYTE = 10 <ENTER>?
                    BEQ       CLOSE       ; IF SO, CLOSE
                    LEA.L     BYTE,A1     ; BYTE ADDRESS IN A1
                    CMPI.B    #32,(A1)    ; IS BYTE = 32 <SPACE>
                    BEQ.S     COPY        ; IF SO, NEW ANOTHER ELLIPSE
                    JSR       ELLIPSE     ; TO REMOVE OLD ELLIPSE
                    JSR       TEST        ; TEST FOR KEY PRESS
.FIRST              JSR       ELLIPSE     ; TO DRAW NEW ELLIPSE
                    BRA       MAIN        ; AND START AGAIN
;
; ***************************************************************
;                          'FBYTE'
;
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
.FBYTE              LEA.L     BYTE,A1     ; BYTE STORE ADDRESS IN A1
                    MOVEQ     #1,D0       ; IO_FBYTE IN D0
                    MOVE.W    #0,D1       ; CLEAR WORD IN D1
                    TRAP      #3
                    MOVE.B    D1,(A1)     ; STORE BYTE
; ---- ADD NEXT LINE BECAUSE THIS IS A SUBROUTINE ----
                    RTS
```

A 'MAIN' routine will normally consist entirely of two things.

1. A set of comparisons of bytes of memory with particular numbers 9i.e. CMPI.B commands), the result of which will dictate the next step.
2. A set of JSR commands to jump to names subroutines.

With a little practice, these 'MAIN' routines can be written easily, and with great complexity.

Listing 2 shows the 'MAIN' routine for this program. The subroutine 'TEST' is a combination of 'UPDOWNFP' and 'UPDOWNINT' to alter the various ellipse parameters according to the key press detected. More about that in a moment. We have given the label 'FIRST' to the second 'JSR ELLIPSE' because we actually start from that point the first time round to draw the default ellipse.

The structure of the 'TEST' subroutine requires a little careful thought in order to produce the most desirable result. You should recall that the 'ELLIPSE' routine requires each parameter as a pair of word integers. The actual floating point para-meter is produced by converting the two integers to floating point numbers, and then dividing the first by the second. The routine 'UPDOWNFP' is designed to produce just such a pair of integers. However, it is also designed to maintain a fix number of significant figures in the floating point number generated. This is ideal for altering the radius, because it means small values change slowly, and big values change more rapidly.

But it would be rather silly to use this to alter the position of the ellipse, because it would mean positions would change slowly near the graphics origin, and more rapidly farther away. Clearly, the distance from the graphics origin is irrelevent to how fast we might wish to move it. So for moving the ellipse, we use 'UPDOWNINT' to alter the first integar at a constant rate, and fix the second one with a suitable default value. A similar argument goes for the angle of the ellipse, so 'UPDOWNINT' is use for that one as well.

Eccentricity

Altering the eccentricity is not so clear cut. A case can be made for altering the eccentricity slowly when it is nearly circular, and more rapidly when it is more eccentric. However, I have opted for altering it at a constant rate using 'UPDOWNINT'.

There are a lot of alterations needed to 'UPDOWNFP' and 'UPDOWNINT' to make them fit our program, but they are very obvious, and repetitive, and, with care, should cause few problems. Clearly, we need to use different key combinations to change different ellipse parameters, and this must be reflected in the values of 'BYTE' we test for at various points. I have chosen to use <LEFT>, <RIGHT>, <UP> AND <DOWN> to alter the ellipse position, <CTRL><UP> and <CTRL><DOWN> to alter the radius, <ALT><UP> and

```
;
.BYTE               DC.B      0
;
; ************************************************************************
;                                 'ELLIPSE'
;
; PUT R1 STACK SO BELOW TOP OF DATA SPACE.
.ELLIPSE            LEA.L     -50(A5),A1          ; RI STACK ON A1
;
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    EXA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0     ; #RI_FLOAT ON D0
                    MOVE.W    $11C,A3   ; RI_EXEC ON A3
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    EXB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE X COORD ON RI STACK.
                    MOVEQ     #$10,D0   ; #RI_DIV ON D0
                    JSR       (A3)
; ---- DELETE LINE TO NEGATE F.P. PARAMETER ----
;
; NOW REPEAT THE WHOLE THING FOR THE Y COORDINATE
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    EYA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0     ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    EYB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE Y COORD ON RI STACK.
                    MOVEQ     #$10,D0   ; #RI_DIV ON D0
                    JSR       (A3)
; ---- DELETE LINE TO NEGATE F.P. PARAMETER ----
;
; NOW REPEAT THE WHOLE THING FOR THE ECCENTRICITY
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    ECCA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0     ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    ECCB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE ECCEN. ON RI STACK.
                    MOVEQ     #$10,D0   ; #RI_DIV ON D0
                    JSR       (A3)
;
; NOW REPEAT THE WHOLE THING FOR THE RADIUS
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    RADA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0     ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    RADB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE RADIUS ON RI STACK.
                    MOVEQ     #$10,D0   ; #RI_DIV ON D0
                    JSR       (A3)
;
; NOW REPEAT THE WHOLE THING FOR THE ANGLE OF ROTATION
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    ROTA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0     ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    ROTB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE ROTATION ON RI STACK.
                    MOVEQ     #$10,D0   ; #RI_DIV ON D0
                    JSR       (A3)
; ---- DELETE LINE TO NEGATE F.P. PARAMETER ----
;
; NOW PLOT THE ELLIPSE
; ---- DELETE LINES TO ENTER ID IN A0 AND TIMEOUT IN D3 ----
                    MOVEQ     #$33,D0   ; #SD.ELIPS ON D0
                    ADDA.L    A6,A1     ; MAKE A1 STACK ABSOLUTE
                    TRAP      #3
; ---- ADD NEXT LINE BECAUSE IT IS A SUBROUTINE ----
                    RTS
;
; INTEGER TABLE
; ---- ALTER TO REQUIRED INITIAL VALUES ----
.EXA                DC.W      80        ; 1ST INTEGER OF CENTRE X
.EXB                DC.W      1         ; 2ND INTEGER OF CENTRE X
.EYA                DC.W      50        ; 1ST INTEGER OF CENTRE Y
.EYB                DC.W      1         ; 2ND INTEGER OF CENTRE Y
.ECCA               DC.W      10        ; 1ST INTEGER OF ECCENTRICITY
.ECCB               DC.W      10        ; 2ND INTEGER OF ECCENTRICITY
```

<ALT><DOWN> to alter the eccentricity, and finally <ALT><CTRL><UP> and <ALT><CTRL><DOWN> to alter the angle. You can easily change them if you wish. Since we are testing for the <ENTER> key in the 'MAIN' routine, this can be deleted from all parts of 'TEST'. And, of course, we have to make suitable changes to labels.

When, in part 5 of this series, I covered 'ELLIPSE' and related routines, I mentioned a problem with negative parameters, a problem I was unable to account for, but able to get round. When I was putting this sample routine together, I found that the problem apparently did not occur. The program works exactly as expected without any special efforts to deal with negative parameters. As a result, I have deleted all parts of the code specifically to deal with negative parameters.

Put Together

We are now ready to put together the chunks of code to make up the program. As before, I shall put this as a series of statements, rather than a flow chart.

1. First, we start in the usual way with 'JOBSTART', with an appropriate job name.
2. We need a suitable window to draw our ellipse, so we use 'CONSOLE'.
3. We must enable a cursor in the window with 'CURSOR'.
4. We must put the window in XOR mode with 'WRITEMODE'.
5. The default values of the ellipse parameters must be copied to the ellipse parameter table with 'COPY'.
6. Next we come to the body of the program, which is our specially written 'MAIN' routine.

Somewhere in all this we must put our subroutines. They don't have to be in any particular place, or in any particular order. In the program listing, I have put them after the 'MAIN' routine, but before the code to close the program.

7. The most important subroutine is 'ELLIPSE', and it must have a duplicate set of parameters.
8. The most complicated part of all is the construction of the subroutine 'TEST' which will test for particular key presses and alter ellipse parameters accordingly. It is constructed as follows.

8a. 'UPDOWNFP' is used to adjust the radius parameters RADA and RADB, and must be altered to test for <CTRL><UP> and <CTRL><DOWN>.

8b. 'UPDOWNINT' is used to adjust the centre Y parameter EYA, and must test for <UP> and <DOWN>. This is the default setting, and needs no alteration.

8c. 'UPDOWNINT' is used to adjust the centre X parameter EXA, and must be altered to test for <LEFT> and <RIGHT>.

8d. 'UPDOWNINT' is used to adjust the eccentricity parameter ECCA, and must be altered to test for <ALT><UP> and <ALT><DOWN>.

```
.RADA                DC.W      20          ; 1ST INTEGER OF RADIUS
.RADB                DC.W      1           ; 2ND INTEGER OF RADIUS
.ROTA                DC.W      31          ; 1ST INTEGER OF ROTATION
.ROTB                DC.W      10          ; 2ND INTEGER OF ROTATION
;
; ---- COPY OF DEFAULT VALUES OF INTEGER TABLE WITH LABEL .DEFAULTS ----
.DEFAULTS
                     DC.W      80          ; 1ST INTEGER OF CENTRE X
                     DC.W      1           ; 2ND INTEGER OF CENTRE X
                     DC.W      50          ; 1ST INTEGER OF CENTRE Y
                     DC.W      1           ; 2ND INTEGER OF CENTRE Y
                     DC.W      10          ; 1ST INTEGER OF ECCENTRICITY
                     DC.W      10          ; 2ND INTEGER OF ECCENTRICITY
                     DC.W      20          ; 1ST INTEGER OF RADIUS
                     DC.W      1           ; 2ND INTEGER OF RADIUS
                     DC.W      31          ; 1ST INTEGER OF ROTATION
                     DC.W      10          ; 2ND INTEGER OF ROTATION
;
; ---- DELETE SUBROUTINE TO NEGATE F.P. NUMBER ----
;
; ********************************************************************
;                          'UPDOWNFP'
; ---- DELETE LINE WITH UNWANTED .BYTE ----
; ****** ALTER RADIUS WITH <CTLR><UP> & <CTRL><DOWN>********
; ---- ALTER LABEL TO .TEST ----
; ---- DELETE LINES TO CLEAR MFLAG ----
.TEST                LEA.L     BYTE,A1     ; BYTE ADDRESS IN A1
; ---- ALTER NEXT LINE TO TEST FOR <CTRL><UP> ----
                     CMPI.B    #210,(A1)   ; IS BYTE = <CTRL><UP> ?
                     BEQ.S     UPFP        ; IF SO, THEN UP
                     LEA.L     BYTE,A1     ; BYTE ADDRESS IN A1
; ---- ALTER NEXT LINE TO TEST FOR <CTRL><DOWN> ----
                     CMPI.B    #218,(A1)   ; IS BYTE = <CTRL><DOWN> ?
                     BEQ.S     DOWNFP      ; IF SO THEN DOWN
; ---- DELETE 6 LINES DEALING WITH <MINUS> AND <ENTER> KEYS ----
; ---- ALTER NEXT LINE TO BRANCH TO UPDOWN ----
                     BRA       UPDOWN      ; IF NEITHER, SKIP TO NEXT BIT
; MAKE NUMBER BIGGER. ADJUST VALUES TO SUIT.
; ---- ALTER LOAD ADDRESS TO RADA ----
.UPFP                LEA.L     RADA,A1     ; RADA ADDRESS IN A1
; ---- ALTER LOAD ADDRESS TO RADB ----
                     LEA.L     RADB,A2     ; RADB ADDRESS IN A2
                     MOVE.W    (A1),D0     ; RADA IN D0
                     CMPI.W    #100,D0     ; RADA ALREADY 100 ?
                     BEQ       NEXTBIT     ; IF SO LEAVE UNCHANGED
                     ADDI.W    #1,(A1)     ; ADD 1 TO RADA
                     MOVE.W    (A1),D0     ; RADA TO D0
                     CMPI.W    #100,D0     ; IS RADA =100 ?
                     BNE       NEXTBIT     ; IF NOT, NEXT BIT
                     MOVE.W    (A2),D0     ; RADB TO D0
                     CMPI.W    #1,D0       ; IS RADB = 1 ?
                     BEQ       NEXTBIT     ; THEN LEAVE RADA,RADB UNCHANGED
                     MOVE.W    #10,(A1)    ; ELSE, RADA = 10
                     MOVE.W    (A2),D0     ; RADB IN D0
                     MOVE.W    #10,D1      ; 10 IN D1
                     DIVU      D1,D0       ; RADB/10
                     MOVE.W    D0,(A2)     ; NEW D0 VALUE IN RADB
                     BRA       NEXTBIT     ; SKIP TO NEXT BIT
; MAKE NUMBER SMALLER. ADJUST VALUES TO SUIT.
; ---- ALTER LOAD ADDRESS TO RADA ----
.DOWNFP              LEA.L     RADA,A1     ; RADA ADDRESS IN A1
; ---- ALTER LOAD ADDRESS TO RADB ----
                     LEA.L     RADB,A2     ; RADB ADDRESS IN A2
                     MOVE.W    (A1),D0     ; RADA TO D0
                     CMPI.W    #9,D0       ; RADA ALREADY 9 ?
                     BEQ       NEXTBIT     ; IF SO, LEAVE UNCHANGED
                     SUBI.W    #1,(A1)     ; SUBTRACT 1 FROM RADA
                     MOVE.W    (A1),D0     ; RADA TO D0
                     CMPI.W    #9,D0       ; IS RADA = 9 ?
                     BNE       NEXTBIT     ; IF NOT, NEXT BIT
                     MOVE.W    (A2),D0     ; RADB TO D0
                     CMPI.W    #10000,D0   ; IS RADB = 10000 IF DIVIDING
                     BEQ       NEXTBIT     ; THEN LEAVE RADA,RADB UNCHANGED
                     MOVE.W    #99,(A1)    ; ELSE, RADA = 99
                     MOVE.W    (A2),D0     ; RADB IN D0
                     MOVE.W    #10,D1      ; 10 IN D1
                     MULU      D1,D0       ; RADB*10
                     MOVE.W    D0,(A2)     ; NEW D0 VALUE INTO RADB
                     BRA       NEXTBIT
; ---- DELETE LINES FOR UNWANTED .SFLAG, .INT1 AND .INT2 ----
; ********************************************************************
;                          'UPDOWNINT'
;
; ************** ALTER CENTRE Y WITH <UP> AND <DOWN> *************
;
; ---- DELETE LINE FOR UNWANTED .BYTE ----
.UPDOWN              LEA.L     BYTE,A1     ; BYTE ADDRESS IN A1
                     CMPI.B    #208,(A1)   ; IS BYTE = 208 ?
                     BEQ.S     UP          ; IF SO, THEN UP
                     LEA.L     BYTE,A1     ; BYTE ADDRESS IN A1
                     CMPI.B    #216,(A1)   ; IS BYTE = 216 ?
                     BEQ.S     DOWN        ; IF SO THEN DOWN
; ---- DELETE LINES TO TEST FOR <ENTER> ----
; ---- ALTER BRANCH ADDRESS TO UPDOWN2 ----
                     BRA       UPDOWN2     ; IF NEITHER, SKIP TO UPDOWN2
; ---- ALTER LOAD ADDRESS EYA ----
.UP                  LEA.L     EYA,A1      ; INTEGER ADDRESS IN A1
                     ADDI.W    #1,(A1)     ; ADD 1 TO INTEGER
                     BRA       NEXTBIT     ; SKIP DOWN
; ---- ALTER LOAD ADDRESS TO EYA ----
.DOWN                LEA.L     EYA,A1      ; INTEGER ADDRESS IN A1
                     SUBI.W    #1,(A1)     ; SUBTRACT 1 FROM INTEGER
                     BRA       NEXTBIT
;
; ---- DELETE LINE FOR UNWANTED .INT ----
; ********************************************************************
;                          'UPDOWNINT'
;
; ********* ALTER CENTRE X WITH <LEFT> AND <RIGHT> *****************
```

8e. 'UPDOWNINT' is used to adjust the angle parameter ROTA, and must be altered to test for <ALT><CTRL><UP> and <ALT><CTRL><DOWN>.

9. We end the program with 'CLOSE' to close the window, and 'ENDJOB' to kill the job.

Putting all that together, we get:

JOBSTART, CONSOLE, CURSOR, WRITEMODE, COPY, MAIN, ELLIPSE, UPDOWNFP, UPDOWNINT, UPDOWNINT, UPDOWNINT, CLOSE, ENDJOB.

Listing three shows the result of merging the required chunks of code into a file and making the approriate alterations to make them fit together. As before, most of the comments have been removed, with lines of asterisks left between the chunks to show clearly where one ends and the next begins. Also, additional comments have been added to point out the necessary alterations. You could, of course, simply copy **Listing one**, and assemble it, but it wouldde feat the point of the exercise. It is much more informative to start from the chunks of code, merge them together, and then make the alterations, and note why those alterations are made. Actually, if you have kept up with this series, you have already typed out the library of chunks of code, it is almost certainly a lot less work doing it this way.

Having got the program to work, it would certainly be worth while experimenting with it, because that will undoubtebly lead to a better understanding of how it works, and, ultimately, to how to write similar routines, customised to your own needs. The first thing it would be worth trying is simply to change the key presses needed to make the various adjustments to the ellipse. You may find one more of your liking. Next, you could alter the speed of adjusting the ellipse by altering the ADDI.W and SUBI.W commands in the 'UPDOWNFP' and UPDOWNINT' chunks of code.

Further Ideas

When you feel a bit more confident with the program, you could add some extras, like using 'FLOOD' to fill in the ellipse before going on to the next. To fill the ellipse you would need to rewrite the 'MAIN' routine so that when the SPACE or ENTER key was detected, instead of going straight to 'COPY' or 'CLOSE', the program would switch on the graphics 'fill', redraw the ellipse, and then switch 'fill' off again.

After all that, you could try applying the ideas to drawing arcs, and 'rubber-banding' lines, and then incorporating them all into the same program. If you then open another window to display a set of cursor commands for each type of figure to be drawn, before you know where you are, you might have written a program for producing artwork on the screen. **Happy coding!**

```
; ---- DELETE LINE FOR UNWANTED .BYTE ----
; ---- ALTER LABEL TO .UPDOWN4 ----
.UPDOWN4            LEA.L     BYTE,A1    ; BYTE ADDRESS IN A1
; ---- ALTER NEXT LINE TO TEST FOR <ALT><CTRL><UP> ----
                    CMPI.B    #211,(A1) ; IS BYTE = <ALT><CTRL><UP> ?
; ---- ALTER BRANCH ADDRESS TO UP4 ----
                    BEQ.S     UP4        ; IF SO, THEN UP
                    LEA.L     BYTE,A1    ; BYTE ADDRESS IN A1
; ---- ALTER NEXT LINE TO TEST FOR <ALT><CTRL><DOWN> ----
                    CMPI.B    #219,(A1) ; IS BYTE = <ALT><CTRL><DOWN> ?
; ---- ALTER BRANCH ADDRESS TO DOWN4 ----
                    BEQ.S     DOWN4      ; IF SO, THEN DOWN
; ---- DELETE LINES TO TEST FOR <ENTER> ----
                    BRA       NEXTBIT    ; IF NEITHER, SKIP TO NEXT BIT
; ---- ALTER LABEL TO .UP4 ----
; ---- ALTER LOAD ADDRESS TO ROTA ----
.UP4                LEA.L     ROTA,A1    ; INTEGER ADDRESS IN A1
                    ADDI.W    #1,(A1)    ; ADD 1 TO INTEGER
                    BRA       NEXTBIT    ; SKIP DOWN
; ---- ALTER LABEL TO .DOWN4 ----
; ---- ALTER LOAD ADDRESS TO ROTA ----
.DOWN4              LEA.L     ROTA,A1    ; INTEGER ADDRESS IN A1
                    SUBI.W    #1,(A1)    ; SUBTRACT 1 FROM INTEGER
;
; ---- DELETE LINE FOR UNWANTED .INT ----
; ---- ADD NEXT LINE BECAUSE IT IS A SUBROUTINE ----
.NEXTBIT            RTS
; ***************************************************************************
;                                 'CLOSE'
;
.CLOSE              MOVE.L    (A7)+,A0   ; CHANNEL ID IN A0
                    MOVEQ     #2,D0      ; #IO_CLOSE IN D0
                    TRAP      #2         ;
;
; ***************************************************************************
;                                 'ENDJOB'
;
.END_JOB            MOVEQ     #5,D0      ; #MT_FRJOB IN D0
                    MOVEQ     #-1,D1     ; ID OF THIS JOB IN D1
                    TRAP      #1         ;
;
; ***************************************************************************
;
; ---- DELETE LINE FOR UNWANTED .BYTE ----
; ---- ALTER LABEL TO .UPDOWN2 ----
.UPDOWN2            LEA.L     BYTE,A1    ; BYTE ADDRESS IN A1
; ---- ALTER NEXT LINE TO TEST FOR <RIGHT> ----
                    CMPI.B    #200,(A1) ; IS BYTE = <RIGHT>?
; ---- ALTER BRANCH ADDRESS TO UP2 ----
                    BEQ.S     UP2        ; IF SO, THEN UP
                    LEA.L     BYTE,A1    ; BYTE ADDRESS IN A1
; ---- ALTER NEXT LINE TO TEST FOR <LEFT> ----
                    CMPI.B    #192,(A1) ; IS BYTE = <LEFT> ?
; ---- ALTER BRANCH ADDRESS TO DOWN2 ----
                    BEQ.S     DOWN2      ; IF SO THEN DOWN
; ---- DELETE LINES TO TEST FOR <ENTER> ----
; ---- ALTER BRANCH ADDRESS TO UPDOWN3 ----
                    BRA.S     UPDOWN3    ; IF NEITHER, SKIP TO UPDOWN3
; ---- ALTER LABEL TO .UP2 ----
; ---- ALTER LOAD ADDRESS TO EXA ----
.UP2                LEA.L     EXA,A1     ; INTEGER ADDRESS IN A1
                    ADDI.W    #1,(A1)    ; ADD 1 TO INTEGER
                    BRA       NEXTBIT    ; SKIP DOWN
; ---- ALTER LABEL TO .DOWN2 ----
; ---- ALTER LOAD ADDRESS TO EXA ----
.DOWN2              LEA.L     EXA,A1     ; INTEGER ADDRESS IN A1
                    SUBI.W    #1,(A1)    ; SUBTRACT 1 FROM INTEGER
                    BRA       NEXTBIT
;
; ---- DELETE LINE FOR UNWANTED .INT ----
; ***************************************************************************
;                                 'UPDOWNINT'
;
; ******** ALTER ECCENTRICITY WITH <ALT><UP> & <ALT><DOWN> **********
;
; ---- DELETE LINE FOR UNWANTED .BYTE ----
; ---- ALTER LABEL TO .UPDOWN3 ----
.UPDOWN3            LEA.L     BYTE,A1    ; BYTE ADDRESS IN A1
; ---- ALTER NEXT LINE TO TEST FOR <ALT><UP> ----
                    CMPI.B    #209,(A1) ; IS BYTE = <ALT><UP> ?
; ---- ALTER BRANCH ADDRESS TO UP3 ----
                    BEQ.S     UP3        ; IF SO, THEN UP
                    LEA.L     BYTE,A1    ; BYTE ADDRESS IN A1
; ---- ALTER NEXT LINE TO TEST FOR <ALT><DOWN> ----
                    CMPI.B    #217,(A1) ; IS BYTE = <ALT><DOWN> ?
; ---- ALTER BRANCH ADDRESS TO DOWN3 ----
                    BEQ.S     DOWN3      ; IF SO THEN DOWN
; ---- DELETE LINES TO TEST FOR <ENTER> ----
; ---- ALTER BRANCH ADDRESS TO UPDOWN4 ----
                    BRA.S     UPDOWN4    ; IF NEITHER, SKIP TO UPDOWN4
; ---- ALTER LABEL TO .UP3 ----
; ---- ALTER LOAD ADDRESS TO ECCA ----
.UP3                LEA.L     ECCA,A1    ; INTEGER ADDRESS IN A1
                    ADDI.W    #1,(A1)    ; ADD 1 TO INTEGER
                    BRA       NEXTBIT    ; SKIP DOWN
; ---- ALTER LABEL TO .DOWN3 ----
; ---- ALTER LOAD ADDRESS TO ECCA ----
.DOWN3              LEA.L     ECCA,A1    ; INTEGER ADDRESS IN A1
                    SUBI.W    #1,(A1)    ; SUBTRACT 1 FROM INTEGER
                    BRA       NEXTBIT
;
; ---- DELETE LINE FOR UNWANTED .INT ----
; ***************************************************************************
;                                 'UPDOWNINT'
;
; ******** ALTER ANGLE WITH <ALT><CTRL><UP> & <ALT><CTRL><DOWN> **********
```

DJC

Dilwyn Jones Computing

41 Bro Emrys, Tal-y-Bont,
Bangor, Gwynedd LL57 3YT U.K.
Tel: Bangor (0248) 354023

GRAPHICS

THE PAINTER V4.04 ..... **£25.00**
[F 512K] 100% machine code art and graphics program, by PROGS of Belgium. Multiple screens, can be mouse controlled, selection of fonts, patterns, many facilities.

THE CLIPART **£12.00**
[F 128K] 3 disks full of clipart pictures, for use with most DTP/graphics programs.

QRACTAL **£20.00**
[F 512K] Machine code fractals program. Displays Julia or Mandelbrot sets.

QRACTAL SCREENS ..... **£ 5.00**
[F 128K] A disk full of sample screen pictures generated with the QRactal program. Buy this to see the sample pictures, then ask about a discount when buying QRactal itself!

IMAGE PROCESSOR V2 ..... **£15.00**
[F 256K] Image enhancement, edge detection, zoom and edit, touch up picture generated by digitisers, scanners or other graphics programs. Works with mode 4 or 8 screens.
UPGRADE OLD IMAGE PROCESSOR **£10.00**
(Send proof of purchase of old version)

PD2 CLIPART **£10.00**
[F 128K] Clipart made for PD2 Plus, but available in screen format for use with other DTP/graphics programs if required.

SCREEN SNATCHER **£10.00**
[F IM 128K] 'Grab' screen displays from within other programs that are not able to save their own displays.

TEXT 'N' GRAPHIX **£20.00**
[F 256K] **NEW!** Allows you to insert screens, or part of QL screen pictures, into a Quill or plain text file printout. Mix NLQ text and graphics in the same printout! Currently only for use with 9 pin Epson compatible dot matrix printers - ask about a version for 24 pin printers due soon.

TRANS24 ..... **£10.00**
[F IM 128K] Translates 9 pin graphics printouts into data for a 24 pin printer.

TEXT

BIBLE TEXT DISKS, EDITOR FORMAT **£20.00**
BIBLE TEXT DISKS, QUILL "_DOC" **£20.00**
[F 256K] Text of the King James Bible on Disk. Please state whether you require Editor (plain text) or Quill '_doc' format.

SPELLBOUND **£30.00**
(F 1M 384K) A spelling checker which can check spelling *as you type!* 30,000 word dictionary

SPELLBOUND SPECIAL EDITION **£50.00**
(F 512K) Enganced version of Spellbound with 50,000 word dictionary & many new features.
UPGRADE TO SPELLBOUND S.E. **£30.00**

QUICK POSTERS **£10.00**
(F 2M 128K) Text poster maker, for use with Star NL, XB and LC printers.

ROB ROY BARGAIN PACK **£10.00**
(F 3M 128K) Reviewed in QL WoRld August 1991

DATA BASES

ADDRESS BOOK & LABEL PRINTER **£15.00**
[F 2M R 384K] Store names and addresses and print them out on a variety of label sizes, or print a telephone list, etc.

DATA DESIGN **£50.00**
[F 512K] Superb, fast 100% machine code database by PROGS of Belgium. Programmable via BASIC and machine code interfaces - write your own database applications in BASIC using the DATA Design commands to help you! Runs in pointer environment (supplied)

FLASHBACK **£25.00**
[F 1M 256K] Fast machine code database which is also very easy to use.

FLASHBACK SPECIAL EDITION **£40.00**
[F 2M 512K] Enhanced version of Flashback, new commands, report generator, etc
UPGRADE TO FLASHBACK S.E. **£20.00**
(return Flashback master disk)

QL GENEALOGIST, STANDARD VERSION ... **£19.50**
[F 2M 384K] Family trees and family history database, one of our best selling programs.

QL GENEALOGIST SECOND EDITION **£30.00**
[F 384K] Enhanced version - for details see our advert in QL World December 1991.
UPGRADE TO SECOND EDITION **£12.00**
(return Genealogist master disk)

BUDGET 128K QL GENEALOGIST **£12.00**
[F IM 128K] Cut down version for unexpanded machines.

SUPER DISK INDEXER **£12.00**
[F IM 384K] Catalogue your disk collection as a database of files. Store it, print it, search & find files, make your life easier.

DBEASY **£15.00**
[F 512K] *NEW!* A database front end for Archive, plus a suite of programs called Chaos Busters. Software from EMSoft, U.S.A.

DBPROGS **£15.00**
[F 512K] *NEW!* A collection of Archive utilities and text files to help you learn to program Archive. Software from EMSoft, U.S.A.

DTP

PAGE DESIGNER 2 PLUS **£40.00**
[F 512K] QL Desktop Publishing program, mix text and graphics, make printed pages or posters, use clipart, use graphics from other programs etc. Prints on most printers. Ask for details! (please note: this program has been delayed, apologies to those waiting– please check if available before ordering)

UPGRADE PD2 TO PD2 PLUS **£20.00**
(Send proof of purchase of old PD2)

QL HARDWARE

MINI PROCESS CONTROLLER **£59.95**
Relay switched outputs, controlled via QQL serial port.

SOFTWARE TOOLKIT FOR MPC ABOVE **£9.95**
[F 128K] Control the MPC from BASIC programs more easily with these extensions.

NETWORK PROVER ..... **£ 3.50**
Plugs between network lead and computer to give a visual indication of transmission.

POSTAGE RATES - SEE BELOW

OTHER QL SOFTWARE

PRINTERMASTER ..... **£20.00.**
[F IM 128K] *NEW!* Tame your printer! Popup utility with menu selection of control codes to send to a printer, Want to change fonts, set paper length, margins, set up for listings or wordprocessing etc. without having to refer to the manual and struggle with control codes. Let Printermaster take the strain out of living with your printer.

HOME BUDGET **£20.00**
[F IM 128K] Domestic bills and accounts program, plus a UK Income Tax Calculator.

REMIND-ME **£12.00**
[F IM 128K] Dates and events reminder program - remember about birthdays, licence renewals, etc. Quick and very easy to use.

REMIND-ME PLUS **£20.00**
(F1M 128k) NEW! Enhanced version with ability to schedule twice as many dates, longer event descriptions, etc.
UPGRADE REMIND-ME TO REMIND-ME PLUS ... **£10.00**
Return master disk with order for upgrade

SCREEN ECONOMISER **£10.00**
[F IM 128K] Turns off the QL display after a set number of minutes to protect the screen.

SLOWGOLD ..... **£ 5.00**
[F 128K] Slowdown routine and control panel for software which runs too fast on Gold Card or indeed any QL system.

TASKMASTER **£25.00**
[F IM 384K] Task switching utility. Enables you to conveniently switch between several programs in memory. Calculator, notepad and file handling utility included.

SUPER DISK LABELLER **£10.00**
[F 256K] Print neat labels for your floppy disks listing the filenames on the disk in columns in small print.

THE CAT ..... **£ 5.00**
[F IM 128K] List files on a disk or cartridge to the screen, or print on paper, in columns (optionally sorted in this latest version). Useful and convenient utility.

★ SEE ALSO THE OTHER HALF OF THIS ADVERT ON THE PRECEDING PAGE ★

SOFTWARE POSTAGE: Software sent post-free to UK addresses. Abroad add £1.00 per program for postage and packing.
SUPPLIES POSTAGE: For disks, boxes and stands, add £2.50 for postage to UK addresses, or 10% of order value (minimum postage £3.50) for airmail postage where possible.
PAYMENT: We can accept payment by cash, cheque (in UK Pounds Sterling only, drawn on UK branch of bank or building society), Postal Order, International Postal/Money order, or Eurocheque. We can also accept payment by these credit cards: VISA, ACCESS, MASTERCARD, EUROCARD and also Barclays CONNECT card. Please state your card type, number, expiry date and your address (goods paid for by credit card can only be sent to cardholder's address). Remember to sign your order. Goods remain property of DJC until paid for in full. Our telephone number above has an answering machine for when we are unable to answer in person – don't be shy!

VISA · Access · EUROCARD · Master Card